
Evaluation of the user interface of a web application platform

Anna Westerberg

30 January 2006
Master thesis, 20 credits
Department of computing science
Umeå University, Sweden

Internal supervisor: Lena Palmquist
External supervisor: Tomas Agerberg
Examinator: Per Lindström

Abstract

More and more people realise their need for usability, which causes the product developers to make better use of the available methods for usability assurance. The assigner for this thesis was Teknikhuset AB, who wanted their web application platform, Content Studio, evaluated. The company wanted to know to what extent the design was usable, what the problems with the design were and how these problems could be solved for future versions.

As new types of applications become more common a tailored set of guidelines gets more important. This became obvious during the progress of this work. It was also clear that methods for evaluation are most effective when combined with each other. On its own, a method is not able to find every usability problem in an interface. For this work a heuristic evaluation was performed along with a questionnaire, interviews and informal observations.

The application in focus of the evaluation was found to be very competent, but with some recurring problems. The recommendations on how to solve the problems were grouped into three areas: Cluttered workspace, user assistance and overall consistency. The recommendations are based on the results from a heuristic evaluation, interviews, observations and questionnaires. Together with the presented guidelines they create a good foundation for future development of Content Studio.

Sammanfattning

Fler och fler inser behovet av användbarhet, vilket får produktutvecklarna att bättre ta tillvara de metoder som finns tillgängliga för att höja användbarheten. Uppdragsgivaren för denna rapport var Teknikhuset AB som ville ha sin webbapplikationsplattform, Content Studio, utvärderad. Företaget ville veta hur användbar produkten var, vilka problem som fanns och hur dessa skulle kunna lösas för kommande versioner.

När nya typer av applikationer blir vanligare blir anpassade riktlinjer för användbarhet viktigare. Detta stod klart under fortskridandet med det här arbetet. Det blev också uppenbart att utvärderingsmetoderna är mest effektiva i kombination med varandra. På egen hand är en metod inte kapabel att hitta alla användbarhetsproblem i ett gränssnitt. För detta arbete utfördes en heuristisk utvärdering tillsammans med en enkät, intervjuer och informella observationer.

När det gäller applikationen som utvärderingen gällde, märktes det att den var väldigt kompetent men att den innehöll återkommande problem. Rekommendationerna för hur problemen skulle kunna lösas grupperades i tre områden: Belamrat gränssnitt, hjälp för användarna och genomgående konsekvens. Rekommendationerna baserades på resultaten från en heuristisk utvärdering, intervjuer, observationer och enkäter. Tillsammans med de presenterade riktlinjerna bildar de en bra grund för framtida utveckling av Content Studio.

Table of Contents

1	Introduction	7
2	Problem description	9
2.1	Goal and purpose.....	9
2.2	In-depth study.....	10
2.3	Conditions.....	10
3	Usability	11
3.1	Definitions	12
3.2	Interaction design	13
4	Content Management	15
4.1	Defining Content Management Systems.....	15
4.1.1	The Enterprise CMS	16
4.2	Additional aspects on CMS.....	16
4.3	Content Studio	17
4.3.1	The users.....	19
4.4	Similar systems	20
5	Evaluation	21
5.1	The DECIDE framework.....	21
5.1.1	Determine the overall goals	21
5.1.2	Explore the specific questions to be answered	22
5.1.3	Choose the evaluation paradigm and techniques.....	22
5.1.4	Identify the practical issues that must be addressed.....	24
5.1.5	Decide how to deal with the ethical issues.....	24
5.1.6	Evaluate, interpret and present the data	24
5.2	Asking experts.....	25
5.2.1	Heuristic evaluation	25
5.3	Asking users.....	27
5.3.1	Questionnaires and interviews.....	27
6	Usability Guidelines.....	30
6.1	Principles and guidelines for Content Studio	31
6.1.1	Consistency	32
6.1.2	Minimize users' memory load	33
6.1.3	Feedback	33
6.1.4	User control.....	34
6.1.5	Prevent errors.....	34
6.1.6	Flexibility and efficiency of use	35

6.1.7	Simplicity	35
6.1.8	Help and documentation	36
6.2	Trade-offs	37
7	Results	38
7.1	Concerns	Fel! Bokmärket är inte definierat.
7.2	User input	43
7.2.1	Interviews	43
7.2.2	Observations	44
7.2.3	Survey	44
7.3	Research from other systems	47
7.3.1	EPI Server and SiteVision	48
7.3.2	New versions of Windows and Office	48
7.4	Summary	52
8	Recommendations	53
8.1	Cluttered Workspace	53
8.1.1	Window management	53
8.1.2	Interface flexibility	56
8.2	User assistance	57
8.2.1	Help	57
8.2.2	Documentation	58
8.2.3	Language	58
8.2.4	Templates	59
8.3	Overall consistency	59
8.3.1	Coherent look and feel	59
8.3.2	Consistent feedback	60
8.3.3	Mapping and visibility	60
9	Summary and Conclusions	62
10	Discussion	63
10.1	Future work	64
	Acknowledgement	65
	References	66
	Appendix A	68
	Appendix B	70
	Appendix C	73

1 Introduction

If usability was just common sense, not so many products would be difficult to use. Nevertheless, without any usability work common sense is precisely what companies have to rely on. Good usability is therefore the result of good work by the development team and is a quality that shows when the product is in use. It shows by making the interaction run smoothly and thereby by being unnoticed.

The author's interest in this work comes from an education in interaction technology and design. In particular, user interfaces have been studied, always with a focus on usability in mind. However, usability is not reduced to just the interface. It may be the product's behaviour that is the key, not what is visible. Hence, many products look very appealing but do not work in use [30].

Most products are released several times with new additions or upgrades. Although, the new releases are not often the result of trying to come to terms with problems the users are having with a product. When it does, it is often made on requests from the users, but the design process still does not involve them and is conducted as usually. Not spending money on including usability work from the beginning often means spending more money later on fixing problems, support calls, getting poor reviews and dealing with unhappy customers [8].

When a company's demand on accessible information increases, a tool to improve and make the process easier is often needed. To solve this problem, Teknikhuset AB developed a product called Content Studio, which is a Content Management System (CMS) used as a platform web publishing. The idea is to make it possible for people with limited knowledge in web publishing to maintain, administer and develop a website or an intranet.

Since usability has a big significance for the user's experience of a system, an evaluation of the system was requested by Teknikhuset to measure how well they have succeeded. The task therefore consisted of evaluating the interface of Content Studio, and suggesting how it can be improved when it comes to human-computer interaction (HCI) aspects. This means identifying how well buttons, menus, dialogs, and the overall structure support the users so that they are able to find what they want quickly and easily.

Due to a forthcoming upgrade of the tool, Teknikhuset wanted to have the user interface (UI) analysed from a usability perspective. This was done to shed light on possible problems and deficiencies, but also to confirm which ideas that were already working and to get an idea of possible solutions for future upgrades. Whether or not the system was easy enough to use was figured out through a usability review and user input. This paper describes the discovered problems, including a recommendation to be used for further development.

Today, user experience goals and new research findings have led to sets of guidelines, which are adjusted to fit various interactive products, such as the web [30]. Interface design for web applications such as Content Studio is one of the areas still lacking a

complete set of suitable guidelines. However, two existing sets of design principles can be used to build the foundation on. These are the Graphical User Interface (GUI) design guidelines for traditional software and the web usability guidelines for web sites [40].

The work showed that it is important to develop guidelines for new types of applications as they become more common. At least, a different way of thinking and an understanding of special circumstances can help increase the usability. It also became obvious that evaluation methods are most effective when combined with each other, since no single method successfully finds every usability problem in an interface. When it comes to the application at hand, Content Studio was proved to have many satisfied customers, but there still existed room for improvement. Mainly three areas were focused on for a possible redesign: Cluttered workspace, user assistance, and overall consistency.

2 Problem description

Teknikhuset AB [35] is a consulting and software producing company focusing on content management, e-learning, and application development. It is located in Umeå, but has customers all over Sweden.

Since the year 2000, Teknikhuset has developed and successfully sold a platform for web publishing and development under the name Content Studio. Continuous development of the product is always made, and a big technical upgrade is in the making for the future release of a fifth version. Included in this work is to make an update of the graphical user interface. So far, Content Studio's GUI has not been developed strictly by any specific usability theories or recommendations other than by following the guidelines for Windows supplied by Microsoft.

This work consisted of analysing the user interface of Content Studio, and identifying deficiencies compared to the theories available in the area, including the recommendations Microsoft has made regarding the interface for Office and Windows. It also included an inspection of the competition as well as the new versions of Office and Windows to get an idea of the market trend for Microsoft. Because Microsoft's applications have such big influence on Content Studio, it is affected by the development of their interfaces. The paper will be used as a base for what will be introduced in coming versions of Content Studio.

2.1 Goal and purpose

The goal was to create a foundation for Teknikhuset to stand on when continuing to develop Content Studio, especially for the version 5.0, which will be exposed to a more thorough change. A usability analysis was performed on the interface, to identify deficiencies and assets, and suggest how this can be dealt with to improve the interaction between users and the system. To be able to do this, a thorough review of the tool was made. Apart from present theories on the area, an expert review and user input through questionnaires and interviews formed the base for the evaluation. The recommendation was based on the result from the evaluation and aims to be used as groundwork for future upgrades. Thus, the statement set out to be answered with this thesis was that by performing an evaluation, the usability of the application could be substantially improved.

The need for this work came from the increasing focus on usability and how it can be used as a means to exceed the competition. High usability can be very decisive for a potential customer when choosing between two applications with similar functions. For Teknikhuset, this work was needed to inform them on what they can change in their product to fulfil the users' needs to a greater extent. It was also a way to introduce usability work in the organisation.

2.2 In-depth study

When trying to get a better understanding of Content Studio and its surroundings, the focus initially lay on different ways of evaluation and what might be needed to perform a review as complete as possible. It was soon clear that being able to follow a tailored set of guidelines would help improve the future interface design. Hence, a more thorough study was based on guidelines, both ones from usability experts and the current and future ones from Microsoft.

2.3 Conditions

The evaluation was based on version 4.0 of the program. It was limited to the GUI and any functions shown to be excessive or wrongly designed.

Even though a lot of time was taken to get familiar with the system before conducting the evaluation, it was found to be rather big. There simply was not enough time to encounter every possible usability problem before or during the formal review sessions. Therefore the final recommendation was based on specific encountered examples, but encouraging an overall consistency, thus making similar, undiscovered, problems a part of the solution range. The recommendation was focused on interaction, such as placement of buttons and menus, rather than form and colours.

Previous to this work an online user test was performed by Teknikhuset. The test, however, was limited to the so called easy mode of the application. Therefore, this mode in particular is not discussed in this paper, but the recommendations include any problems discovered there that correspond to areas in normal mode.

The paper was intended to be used as a source for guidance to the company when developing new features and versions. The recommendation itself was not solely based on what currently was possible to introduce, but can be seen as a wish list, leaving it up to Teknikhuset to choose which parts to apply. The guidelines, however, are fairly general for Content Studio as a system, and can be viewed when developing templates for customers as well as new features or dialogs. Adjustment to the way of work, information and context is not completely inherent in the guidelines, since those factors differ for every client and customer.

3 Usability

“Usability means to make sure that something works well: that a person of average (or even below average) ability and experience can use the thing - whether it is a web site or a revolving door – for its intended purpose without getting hopelessly frustrated”
Steve Krug, Don't make me think

Due to the fact that the focus of this work lies in usability aspects and interaction design, a proper introduction to the terms are made in this section. This is done to make distinctively clear what is meant when using them as a goal and a mean for the recommendations, as well as to explain their impact on a system. The chapter also explains the reasons for introducing more usability work in the company's work process and how this can be achieved. The sources come both from the areas of traditional software design and the web, since usability is needed in all types of systems.

People having problems to use a VCR or all the features in a mobile phone are often used as examples to illustrate poor usability. These are problems almost everyone has encountered in one way or another. The field of Human-Computer Interaction has a lot of focus on usability, because it aims to bridge the gap between people and machines. In order for such tools to be effective, it must allow their users to accomplish their tasks in the best way possible. Usability depends on different factors concerning how well the product fits the users' needs, tasks, and expectations. Often usability is considered to be something that every system automatically contains, and only comes into focus when something does not work. The user expects the product to be usable, not realising that it takes work to develop usable interfaces [10]. Guidelines for usability and design, described in a later chapter, can assist designers in creating better user interfaces, but to be able to create an application with high usability, one has to involve some of the people who will actually use the system.

Simply, usability is the quality of a system that makes it pleasing, error tolerant and easy to learn, use and remember [37]. It has nothing to do with whether or not something works. A website or a system can work perfectly well, but be impossible to understand. This is why it is important to understand who the users are, and what knowledge they have [18].

To neglect usability is always a bad choice [29]. It is certainly possible to release a working product without any usability work, but with it, several areas can be improved. For the user usability affects whether a task is performed correctly or not and whether the process was enjoyable or frustrating. The developers' interest in usability should come from its possible impact on the product's success. A software product with poor usability can even manage to decrease the productivity of the

workforce, which is why company management also is affected by it [37]. Tasks that take unnecessarily long to perform, or unnecessary errors made, can create stress and discomfort in the long run. Hence, at all times, lack of usability can cost time and effort.

Many times it is possible to profit from usability investments [24]. If they are able to choose, people will tend to buy systems with a higher degree of usability, which is why usability also can help differentiate the product from competitors. If two products have similar appearance and functions, small differences in usability can have a big impact on users' preferences [8]. Having good usability means that one does not have to invest and develop functionality that is not being used. It helps decrease the time it takes to learn and can increase the earnings through the ability to attract new customers and keeping the present ones. It enables the users to more clearly get an idea of what their possibilities are, which in turn results in fewer calls to the support [18]. As for Teknikhuset, several of the customers are entitled to free support through their licence agreement, which is why the vendor can benefit from less support time. In other cases product support can be very expensive, and charging customers for it can increase the dissatisfaction with the product. The easier the users find the product to be, the less help they will need and the less time and money it will cost for them training. According to Molich [18], the biggest profit comes from having pleased users from the beginning and thereby preventing unexpected problems. Customers that like a product recommend it to others. The perception of a system is directly linked to the vendor, which is why usability quality can be a tool for marketing.

The key principle for reaching an acceptable level of usability is to make use of iterative design, where the design process is repeated continuously and refined and tested for each cycle. The designers are not typical users and neither are developers or usability experts, which is why the real users of the product should be consulted as early as possible. By including user feedback from the beginning, the designers and developers are able to gradually refine the design according to the users' needs. To be able to do this one also has to determine who the primary users are, how they work, and what sort of tasks they must perform. This, together with various methods for evaluation can be used to get the necessary information needed to create an appropriate design. Different evaluation methods are more suitable than others for every project and aspects such as time and money are especially influencing [37]. Krug [12] claims that 80% of mistakes can be caught if a usability expert is brought in from the beginning. Hence, usability should be a part of the design process instead of something that, if possible, is added in the end.

3.1 Definitions

Several definitions of usability exist. Both Nielsen [21] and Molich [18] define the abstract concept of usability as something more measurable by being easy to learn, easy to remember, efficient to use, understandable and satisfying.

The ISO 9241-11 standard [29] gives a definition of usability as “the extent to which a specified product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”.

3.2 Interaction design

A well designed interactive system has an interface that disappears in the background and allows the users to concentrate on their exploration and pleasure. However, creating this sort of environment requires hard work from the designer [32]. If something works in an unexpected way or is unintelligible the interaction comes into focus. Interaction is about collaboration between the user and the computer. In its simplest form, it is just about how to navigate between web pages, while other types of interaction, such as online shopping or searching, might be more difficult. Interaction design is needed not just for web design, but also for things, such as phone systems, thermostats, light switches, and door knobs [33].

When designing for the user, one has to consider how the system will fulfil the users' needs. Preece et al [30], divide the goals for doing that into two categories. The first is goals that deal with how usability specifications can be met, whereas the second deals with how the interaction can feel to the user. The usability goals are:

- **Effectiveness.** How well the system does what it is supposed to.
- **Efficiency.** How the system supports the user when performing a task and how a high productivity is attained once the system is learned.
- **Safety.** How the user is protected from undesirable situations. Unwanted actions should not be possible to accidentally perform.
- **Utility.** How well the users can do what they need and want.
- **Learnability.** How easy the system is to learn. How quickly the users can get started and become competent.
- **Memorability.** How easy the system is to use once it is learned. If users have not used the system in a long time, they should be able to remind how to use it.

These goals are highly suitable for applications that are being updated to improve how work is performed. However, this is not the only relevant factors (see figure 3.1.). How the system is experienced is also an important aspect. This is why one should also try to create systems that are satisfying, enjoyable, fun, entertaining, helpful, motivating, aesthetically pleasing, supportive of creativity, rewarding, emotionally fulfilling.



Figure 3.1. Usability and user experience goals. Inspired from Preece et al [30].

Not every goal is fulfilled in every product, but they allow the designers to use different combinations to reach different users' needs. Although some combinations may be incompatible it is always the specific task, product and user that should determine what is important [30].

Together with graphic design and information design, interaction design is a part of the interface design (see figure 3.2.). For instance, the graphical designer decides colours, shapes and fonts for buttons, while the interaction designer decides whether or not a button is needed in that position, and what will happen if it is pushed. Although, as far as the user is concerned, the entire interface containing shape, information and interaction *is* the application [33].

Most web sites contain all three types of design. At the same time, some sites, such as advertising sites with flash movies, mostly consist of graphical design. Government authorities or organisations focus on information, whereas e-commerce and web applications, such as Content Studio focus on interaction. The evaluation described later in this paper was also more aimed at interaction deficiencies than graphical design. The different types have different focus, but they overlap and in these areas many of the usability problems can arise [33]. Usability problems can be defined as aspects of a user interface that may cause the resulting system to have reduced usability for the end user. A user can find an interface element to be problematic for many reasons: it might make the system harder to learn, it might slow down performance of their tasks, it may cause usage errors, or it might simply be unpleasant [22].

Usability is a feature and an end result, which comes up when each of the designers have performed a good work. User experience is more and more used as a summarizing term for usability, utility and usefulness. Whether or not a product is easy to handle, what it actually does and how useful it is, all contribute to the user's experience. Henceforth, when using the term usability, this will include the user experience but not the term user friendly, which many times is used in this context. Nielsen [11] warns about a too broad use of the term, claiming that people want computers they can use and that do what they want, rather than friendly ones. To make proper use of the term usability and apply the perspective on the task at hand, this sort of wider view was applied, as opposed to just talking about ease of use and fast learning.

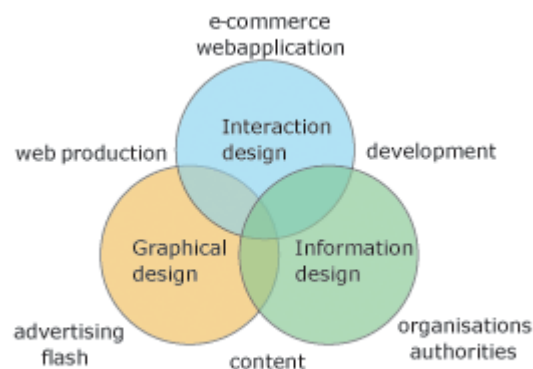


Figure 3.2. Different types of design, inspired from Jonas Söderström's web page [33].

4 Content Management

“A Content Management System defines the way in which content is managed within an overall content management life-cycle from creation to publication”.

Martin White, the Content Management Handbook

At the beginning of the year 2001, the content management industry was still very new. When the web moved into larger, more rapidly changing sites, the need for strong management tools arose. The solution became content management systems (CMSs). Such systems help companies become and stay organized, while their site can grow and change quickly. Today, most of the activity in content management lies within web development, but the web is just one of many information channels that organizations need to manage. As the information sharing between these channels increase so does the need for an organized approach [5].

Most CMS products offer features that allow content to be authored, reviewed, version managed, administered and published. They may offer this in different ways, but one of the biggest benefits is that they make it possible for employees with little or no html knowledge to add content. When content is taken from storage and changed, a new version is created. This makes it important to ensure that no content can be edited by more than one user at a time. The workflow system usually built in the software is there to make sure that content is approved before being published [39].

Content management systems have a simple purpose - to make it possible to get hold of content, to know what content is available, and to make sure that it can be distributed to a variety of places automatically. Yet still no standards are universally accepted for what content management systems are or do [5]. Some existing definitions of the term are therefore presented, to give a better idea of the most common opinions.

4.1 Defining Content Management Systems

There are many definitions of Content Management and its belonging systems, and most of them give a similar description. That is, that Content Management is a general term referring to how information, such as text, images and documents can be organized and structured so they can be stored, published, and edited with ease and flexibility. A content management system is used to simplify content maintenance and design changes, and helps manage content, usually separating it from page design. Authors and editors often need to restore older versions, which is why the ability to manage versions of content is a critical aspect of content management. The belonging software is therefore well suited for content that is regularly updated and published over and over again [9, 19, 38].

CMSs can take several forms, but often the term refers to a web content management system, which generally is a web application for web site management. Parts of the site can consist of HTML files and other parts can be created dynamically out of a database. In a web CMS, all of the following can be included [5]:

- A CMS application, which takes care of collecting content from contributors and managing workflow and administration.
- A repository, which holds all content, administrative data and any of the resources needed to build the site (such as graphics and style sheets).
- A set of HTML files, which the CMS position to the static part of the site.
- A CMS-generated database to which the CMS can set up data and content from its repository.
- Templates, which move data from any source to the state it needs to be in for the site.

4.1.1 The Enterprise CMS

An Enterprise Content Management System (ECMS) integrates a web CMS with additional applications, such as document management, which manages the creation, revision and distribution of formatted documents; records management, which manages the storage and retrieval of created and received documents; and digital asset management, which manages non-text content, such as images and video clips [39].

The potential for an ECMS goes far beyond the web. It can cover the entire content creation and organization system and provide a repository where information can be reviewed and worked on independently of what page it may be used on. Most importantly, it can produce Web sites and any other publication that one wishes to make from the stored content [5]. Based on these definitions, Content Studio can be considered to be more of an enterprise CMS, since it provides additional functionality such as the ones described above.

4.2 Additional aspects on CMS

A proficient implementation of a CMS should enable the organisation to customize information for the users, because the content can be added and stored once, and then reused in different ways. The way the information is structured highly affects how the users are able to navigate on the site. Together with good usability, good information structure enhances the user satisfaction when it enables the users to easily find what they want. Usability testing is essential and not an option, if one wants to achieve this. For the website created with the CMS, usability and accessibility is crucial for the visitors, but it is important for the staff using the CMS authoring interface as well [39]. Hence, usability and accessibility is important factors for the software itself as for the website or other publications created with it.

Implementing a CMS involves many people in an organisation, each with different concerns. Web authors may wonder how long it will take to get used to the

software, and how much training is required, which is why ease of use and an intuitive interface can be crucial when acquiring new customers [39].

4.3 Content Studio

Content Studio is a platform for content management, collaboration, integration and automatization.

Distributing information responsibility throughout the organization usually means that the IT-department's workload can be reduced at the same time as the quality of the information is increased. Content Studio makes it possible for users to contribute information and knowledge without the need for knowing the underlying technology. All administration and editorial work is performed via a standard web browser and Content Studio automatically takes care of all publishing, formatting, and distribution. The database approach of Content Studio allows for information to be reused, edited and controlled by authorization and workflow. More detailed information on Content Studio is available at its website [34], which is where most of the information in the following section was received from.

The platform can be used for handling several parallel websites, in many languages, and in many formats. All site management is handled through a unified interface and in a single installation. For distributed organizations that require sharing of information and templates, Content Studio supports cross-publishing and information can be moved between sites through safe channels.

Content Studio allows information from underlying systems to be imported in order to be published later on different formats such as HTML, XML, PDF and Microsoft Word or Excel. Since information is stored in a XML format, a data format applicable for all types of information, and not limited to a single website, the information can be distributed in various formats to several receivers. XML-technology in Content Studio secures that information is reusable and system-compatible. The use of XML also ensures that information is held separate from its graphical form and structure.

Content Studio consists of 150 compatible components that are used through menus and drag-and-drop for building everything from simple websites to complex systems. The technique for turning advanced program code to reusable objects is called Active Scripting (AS). The developer can create customer suited components that can be reused and upgraded with new functionality. The administrator of the website does not need to have knowledge of the code, and can therefore focus on the need, function and customer benefit. The components in Content Studio can be combined to create modules such as a chat function, forums for discussion and image databases. AS components can be created by using Content Studio, Macromedia Dreamweaver, or other similar programs and can be written in ASP and JavaScript for instance. An application programming interface (API) is also available for those who need to extend the system by traditional programming. The basic version of Content Studio already contains all functions for simple information handling for a classic website or intranet. So, with Content Studio the need for an external consultant decreases, allowing the owner more control over their system.

In short, the core of Content Studio consists of web publishing, but on a deeper level one can find a generic platform for publishing, information

management, integration and automatisisation. The 150 components delivered with the system, collaborates in creating a dynamic and database driven platform for web publishing. Content Studio has been developed for several years and is an established product containing all the functions that can be expected by a web publishing tool.

The application can be used in a so called simple mode (see Figure 4.1). It enables the user to view the documents as they would look when published, but have icons for adding and editing where this is possible. Otherwise, when used in normal mode, it looks like in Figure 4.2.

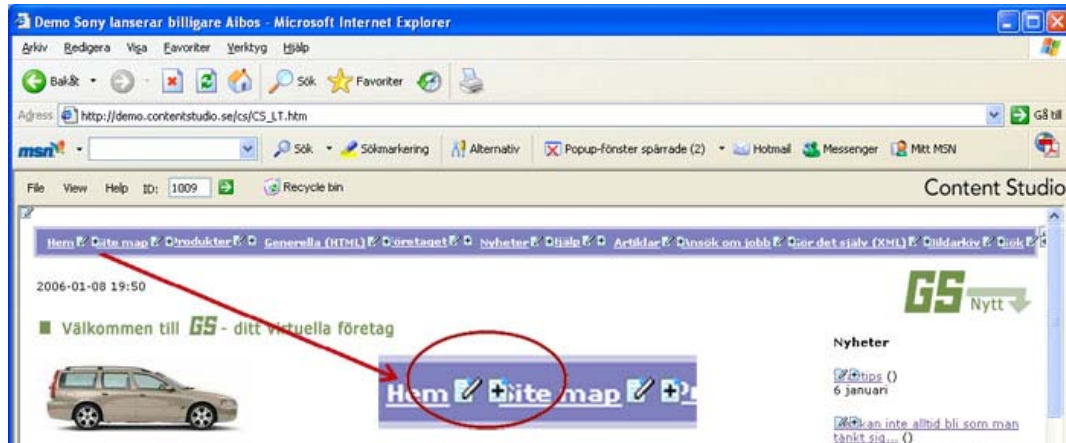


Figure 4.1. The simple mode. The enlargement shows the “edit” and “add” icons.

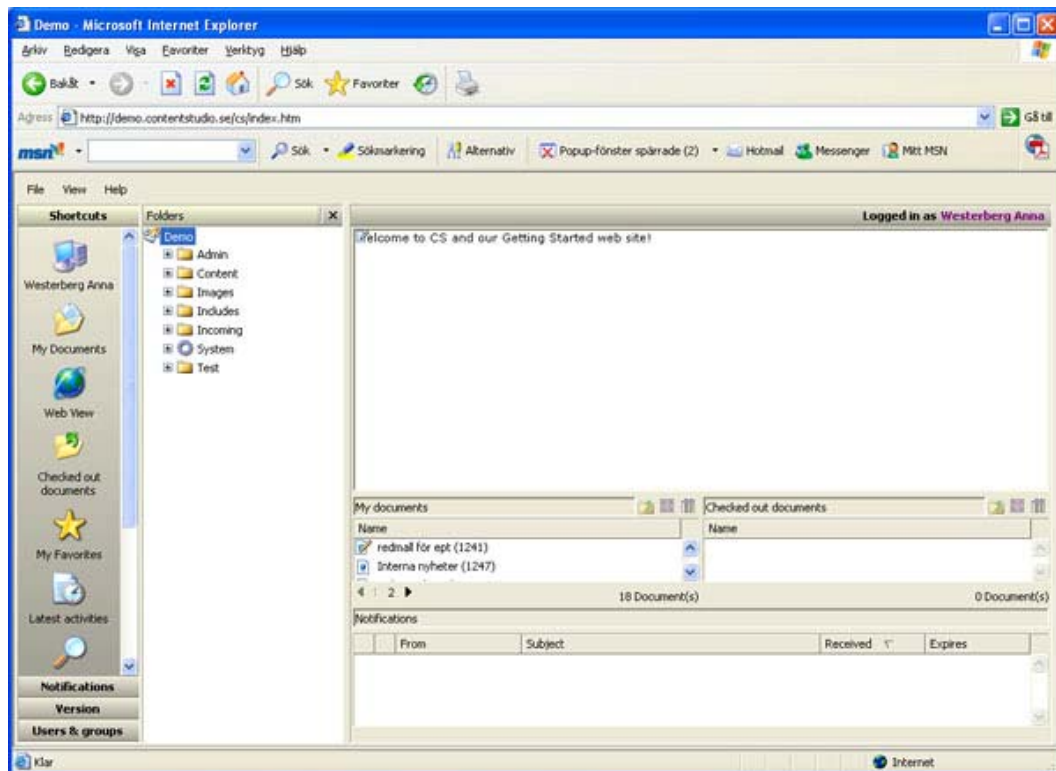


Figure 4.2. The normal mode. From left to right: Accordion buttons, web tree, folder content and file information.

4.3.1 The users

As stated earlier, an important aspect of the design is to know who the typical users of the system are. To understand them, their environment, tasks and organisation is a key to the entire development process. The knowledge includes characteristics such as experience, computer knowledge and system usage, which can be extracted through questionnaires, interviews and observations. Lots of information can also already be present in previously made analyses or other background material, but regardless of method, the analysis aims to determine the demands on the user interface [10].

Today, Content Studio is being used by approximately 100 companies and organisations. Most of them are administrations, energy companies and other private companies with highly trafficked web sites, and also often with other internal and external systems and applications. Among the end users are customer service personnel, informers, product responsables, IT technicians, web designers, etc. The knowledge of the users therefore ranges from very limited to developer.

Three user types have previously been identified by Teknikhuset. These are:

- 1. The editor.**

That is, users of content studio who are writers with no or little experience of web design and mostly will edit information on the site. An editor has rights to create, edit, and delete information, but may not be allowed to publish it. If not, this is usually done by someone who is editor-in-chief or publisher.

- 2. The administrator.**

That is someone who will handle information structures, edit forms and presentation templates, access and backup of the web site. In other words, it is the person that maintains and administers the web site, also known as a web master.

- 3. The developer.**

Someone who develops solutions based on Content Studio.

Besides these user types, writers creating information and visitors using the website can also be acknowledged as roles in the system. Users also have different knowledge and usage patterns. Shneiderman [32] identifies 3 different patterns:

- 1. Novice users**

Have little knowledge of the interface and do or do not know the concept of the task. They need support to overcome their limitations and it is important to make them feel more confident and less intimidated.

- 2. Knowledgeable intermittent users**

These users have knowledge about both the task and the interface concept, but are less likely to remember where functions are located. They need to be assisted with features helping them recognize the structure and how a task should be performed.

- 3. Expert frequent users**

Are very familiar with the concepts and want to be able to work efficiently. This applies to response times, feedback and shortcuts.

When multiple usage classes must be accommodated in one system, the basic strategy is to permit a multi-layer approach to learning. The separation of content and structure in Content Studio enables organisations to redesign a web site without affecting its content. This approach to content management also means that organisations can create web sites for groups of users with particular requirements.

However, at all times it is important to design for exploration. Users cannot be afraid to push buttons not fully remembering what will happen if they do, and must be able to relax. User assistance begins with a good design methodology where one tries to understand the users, how they work, and the problems they encounter. As often as possible, the basic design principles and processes are followed to create an interface that users can learn easily and operate efficiently [15].

4.4 Similar systems

EPiServer and SiteVision are two other content management systems, where EPiServer can be considered to be the biggest competitor of Content Studio. There exist several other similar systems on the market, each of them offering much of the same functionality. Because of this, usability might become the weapon that can convince the customers to buy a particular system.

EPiServer consists of a core containing functionality for editors and administrators. The way of work is pretty much the same for any website made in EPiServer, it is just the website itself that is unique. Both EPiServer and SiteVision are focused on being a WYSIWYG (What-You-See-Is-What-You-Get) editor, where quick access to simple editing is of importance. On paper, all three seem to work very similar in the underlying structure. However, it was the differences in the interface which was of interest for this work. The result of the comparison between them is presented in section 7.3.1.

5 Evaluation

“Evaluating what has been built is very much at the heart of interaction design.
Its focus is on ensuring that the product is usable”.

Preece et al, Interaction Design: beyond human-computer interaction.

As mentioned in chapter 3, several types of evaluation can be used to retrieve the necessary information about the users’ needs and to find deficiencies in the interface. Of course, different evaluation methods are more suitable than others for every project and to determine which these are for this work, the available ones had to be examined.

In the case of an upgrade, there is limited scope for change and attention is focused on improving the overall product. This type of design is well suited to usability engineering in which evaluations compare user performance and attitudes with those for previous versions [32].

5.1 The DECIDE framework

By following the DECIDE framework, presented by Preece et al. [30], a guide for evaluation was provided. The framework helped to put together a plan for evaluation and to identify possible problems and ways to deal with them. The framework consists of six parts, all described in more detail in the following text.

5.1.1 Determine the overall goals

The goal was to identify how the interface of an existing product, Content Studio, could be redesigned and further developed to improve its usability. The evaluation resulted in a list of problems, which was complemented with recommendations of how they could be eliminated in various ways. Both the list of problems (found in Appendix A) and the recommendations were to be used by the company as a help when developing the next version of the application. When doing that, it was desirable that the interface will show that something besides the underlying technique has been updated and improved. When performing the evaluation a list of principles and guidelines was followed. These can be used later on when additional changes or upgrades are made. The goals were determined together with Teknikhuset and adjusted to fit within the specifications and the time constraints of the project. By determining the goals, it becomes more obvious what kind of questions that has to be answered at the same time as it helps to keep focus on the results.

5.1.2 Explore the specific questions to be answered

Of course, there were many questions to be answered when performing the evaluation. Most of them regarded the quality of the interface, such as what the disadvantages with the current system were and why. Not just disadvantages were in focus, although those were the ones that formed the base for the final recommendations. Things that already work in a good way should be kept. However, the main question was if the interface is badly designed in some way. Many of the questions following this concern were about whether or not the users were able to find what they want and need. Was the interface difficult to understand, learn or remember how to use or was it difficult to navigate? Also, the usability guidelines described later in this paper had a big impact on the type of questions asked. For instance, questions about feedback, consistency and language are directly related to the guidelines. With Content Studio being a web application platform, it was also interesting to examine how the integration with the web worked.

5.1.3 Choose the evaluation paradigm and techniques

When the questions were determined, they had to be answered through evaluation. Preece et al [30] have divided the different methods into four main paradigms for evaluation as being:

1. Quick and dirty

When performing a quick and dirty evaluation, one gets feedback from users or consultants regarding the design and how well it fits the users' needs. The main advantage is that the method does not require a lot of time to perform and at the same time it results in suggestions for improvement. When talking to consultants, one can benefit from their technical and user knowledge and their ability to review software quickly. Regardless of who is providing the feedback, it is a very informal method.

2. Usability testing

Usability testing measures how quickly and flawlessly typical users can perform carefully prepared representative tasks. When performing the task, the users are observed in a laboratory environment where everything they do and say is recorded. The information is later used to identify the reasons the users made a mistake trying to perform a task. Questionnaires and interviews are also used to obtain the users' opinions.

3. Field studies

The key element in field studies is the fact that they are performed by studying the users in their natural settings. Qualitative techniques used in field studies are interviews, observation, and ethnography. These techniques are used differently depending on whether one chooses an insider or outsider approach to the study.

4. Predictive evaluation

Predictive evaluations do not need to involve users. Instead it involves experts trying to predict usability problems. To help them in their predictions, the experts have the knowledge about typical users and guidance from heuristics. Due to the fact that

		Evaluation Paradigms			
		Quick and Dirty	Usability testing	Field Studies	Predictive
Techniques	Observing Users	X	X	X	N/A
	Asking users	X	X	X	N/A
	Asking experts	X	N/A	N/A	X
	User testing	N/A	X	N/A	N/A
	Modeling users' task performance	N/A	N/A	N/A	X

Figure 5.1. The relationship between paradigms and techniques for evaluation [30].

users do not need to participate, the method is quick and rather inexpensive. At the same time it has some limitations: for example that the wrong set of heuristics can lead the designer the wrong way.

The techniques extracted from these paradigms are used in various ways depending on the chosen evaluation paradigm. Figure 5.1 shows the relationship between the evaluation paradigms and the different techniques. The techniques are:

Observing users

Challenges are how one observes without disturbing the users and how to analyze the collected data. The common ways of recording the observations are notes, audio or video.

Asking users their opinions

Interviews and questionnaires are the main techniques used to find out what the users think about a product. The number of users and how the questions are structured may vary.

Asking experts their opinions

Experts evaluate the usability by playing the role of a typical user, trying to find possible problems and their solutions.

User testing

Users' performances are tested and analyzed in a controlled environment.

Modelling users' task performance to predict the efficiency of a UI

One wants to test different designs without building complex prototypes. Instead different techniques, such as GOMS (goals, operators, methods and selection rules) and the keystroke model, are used to model human-computer interaction.

Usability testing, and user testing in particular, would maybe have been the most appropriate method, but due to the time constraint and the limited access to users, this was not selected as the main paradigm. Instead the predictive evaluation was chosen. Usability testing *was* used, but only through the technique where the users are asked for their opinions. This was made through questionnaires and interviews. Both of these methods are described in detail in the following sections.

Before the users were asked to provide feedback, the predictive evaluation was performed. One of the two techniques available, asking experts, was used (see figure 5.1). The expert used to perform the evaluation was the author of this paper. The

other technique, modelling the users' performance by using models such as GOMS, was not applied. This decision was based on the author's previous experiences where GOMS was not suitable for evaluating this type of applications. Statements that it relies on small, routine tasks and thus has limited scope [30], also had an impact on this decision. A flexible system such as Content Studio makes it difficult to predict an average user's performance of a task and thereby makes it harder to apply predictive models. It also has long and sometimes advanced tasks that can be performed in many different ways, which further increases the complexity of predicting the performance.

Besides formal questionnaires and interviews, several informal conversations were made with a number of users. This falls under the quick and dirty category, where the discussions with existing and potential users were made both individually and in groups. The observations were not completely made in the users' natural environment, but could otherwise have qualified as field studies.

A minor web-based test with novice users, where the users performed different tasks and answered task related questions during the test, were already in progress at the beginning of the work with this paper. The test was therefore not being conducted as a part of this work, but any result of interest was taken into consideration when putting together the recommendations.

5.1.4 Identify the practical issues that must be addressed

For the online usability test performed by Teknikhuset, only novices were used.

The questionnaires were handed out to as many users as possible, reaching both men and women of various ages with different degrees of knowledge and experience. The users answering the form were recommended by persons well familiar with the client population. To not have to deal with paper handouts when performing the survey, the questionnaire was published as an online form. No particular equipment was needed, except for a computer and a possibility to distribute the form online. There was no budget for this project either.

A time constraint forced the evaluation to be limited to the predictive paradigm and to focus on user opinions instead of formal observations.

5.1.5 Decide how to deal with the ethical issues

To prevent data collected about the participants to be associated with specific persons, no names appear in the report. The form explained the aim of the test and ensured the participants that their performance and personal details would not be made public. Other details they were informed of were for example the estimated time and how the data would be handled and analyzed. They were also told that their performance was to be taken seriously and that it would help improve the next version of the system, and in turn their own work situation.

5.1.6 Evaluate, interpret and present the data

Depending on how many users participating in the tests, the reliability can vary. Part of the evaluation was informal, and therefore had lower reliability. This includes the

observations and some interviews. On the other hand, the web-based test is easy to copy and the more users participating, the more reliable the average results becomes. Since the goal was to find the problems in the current interface, having had experts, frequent users and novices performing some sort of evaluation has hopefully given an overall picture and discovered the most immediate problems.

Possible biases were too few expert reviewers, perhaps with too little experience of everyday use. Not being able to watch neither the users in their natural environment nor the users performing the web-based test may have resulted in many aspects never being considered. The way questions were phrased may have influenced the users when answering them.

Having all sorts of users present in the test, gave the evaluation the scope needed to be able to generalize the results. When providing the expert users with questions, they may not be able to remember problems they may have had in the past and are now familiar with and have a solution to. On the other hand the novice users might pick up more of those kinds of problems and miss out on the ones not being discovered until having to repeat them several times a day.

5.2 Asking experts

When users are not easily accessible, or it is too expensive or time consuming to involve them, experts can perform a review of the system and provide feedback. This can be used in any stage in the design cycle and does not require a lot of resources.

Before evaluating Content Studio by applying guidelines, first a trial run of the system was made while initial impressions were recorded. This was made because of the risk that learning how the system works might make it harder to be able to notice some of these initial problems [6]. Second, a lot of time was spent using the system to be able to reach a higher level of knowledge and thereby finding even more problems once performing the official evaluation.

5.2.1 Heuristic evaluation

Heuristic evaluation is a popular usability inspection method developed by Jakob Nielsen and his colleagues. It is an informal technique where experts are guided by a set of usability guidelines to evaluate usability related aspects of a user interface. When the guidelines are used in the context of an evaluation they are called heuristics. The evaluation is performed to check if the system or product complies with the design principles. The goal is to find the usability problems, compare them with the heuristics and point out where the system fails. In addition to the checklist, the evaluator is also allowed to consider any relevant additional usability principle that may come to mind [22].

As the quotation below shows, heuristic evaluation (also known as usability inspection), was well suited for the aim of this paper.

“Typically, a usability inspection is aimed at finding usability problems in an existing user interface design, and then using these problems to make recommendations for fixing the problems and improving the usability of the

design. This means that usability inspections are normally used at the stage in the usability engineering lifecycle when a user interface design has been generated and its usability (and utility) for users needs to be evaluated [22].”

The guidelines used to perform a heuristic evaluation are general rules that describe common characteristics of usable interfaces. If needed, it is possible to develop specific heuristics that apply to a specific class of products as a supplement to the general principles. The guidelines used for this work are described in chapter 6.

During the evaluation, the experts take the users role and note all the problems encountered. Nielsen recommends that the best number of experts is 3-5 [22]. Additional experts do not tend to discover more problems. The number of experts used often depends on how much money that can be saved if the misuse of the system is limited to a minimum [21]. One expert finds, on the average, 35% of the usability problems of the interface, but of course there are individual differences. The best result comes from using evaluators that are both experts on usability *and* the system in question. Previous testing [20] has shown that these find on average 60% of the problems. Usability experts find 41% of the problems and novices find 22%. However, increasing the number of evaluators to five caused the percentages to reach 98%, 80% and 50% respectively.

The problems with heuristic evaluations are that knowledge about the users, their tasks and the context of use is crucial for the quality of the evaluation. There is also a possibility that the HCI knowledge of the expert may be too low, due to lack of experience or education [22]. It is, however, a quick and cheap way to discover fundamental usability problems, but it may not deliver complete insight of the problem [10]. Although finding many usability problems, the experts may miss some problems that can be found by user testing. If the system highly depends on the domain and the experts have little domain expertise they are likely to overlook some usability problems. In such cases it is almost impossible to find the problems without user testing. The best way according to Nielsen [22] is to use both methods. For instance, first perform a heuristic evaluation to eliminate the most obvious problems and then, when the design has been improved, let the users test it to find whatever problems remain. According to Barnum [1] heuristic evaluation is a good predictor of end-user problems, but no method is as good as actual user testing. Typically, these methods do not result in the same findings, so again it is said that it is best to use several methods to determine the usability of a product.

For this work, the heuristic evaluation was performed first to get an as objective review as possible. Following Nielsen’s [22] recommendation the user interface for Content Studio was reviewed several times. This was needed to get a better understanding of the system and how it works, as well as to reach a higher level of usage to be able to encounter more problems. Every impression and problem was noted at first and at a later occasion, more specific elements were evaluated when knowing how they fitted into the bigger context. Every training tool available, such as tutorials and manuals was taken advantage of, as recommended by Shneiderman [32].

The result from a heuristic evaluation is, as mentioned, often a list of usability problems, each of which have references to the specific guideline that is violated and why. This was also the result of this work, but it also includes recommendations and possible solutions to these problems, all of which are described in the following sections. This type of evaluation often does not deal with the positive sides of an

interface, but this is addressed in this paper. Since no guarantees can be given that the results of a review will find every problem in an interface, supplements with other methods was made to get a more reliable evaluation. Heuristic evaluation and usability testing complement each other nicely according to Dumas and Redish [8]. Following this recommendation, the predicted problems and concerns from the expert review was used as a foundation for the questionnaire.

After conducting the heuristic evaluation the users were asked to provide their opinions in order to draw attention to any undiscovered problems. The results from the two methods were compared to see where they overlapped and where problems not found by the evaluator were located. During the period of user input, informal observations and interviews were made on several occasions. Some of these observations could fall under field studies, which according to Barnum [1] are usually performed on systems already in use and often gives very relevant results.

5.3 Asking users

Access to the users was provided through courses called developer days held by Teknikhuset, where their clients come to learn about new features in the application or similar topics. At each occasion 8-10 users were available for observation and discussion. Altogether, four such occasions was taken advantage of.

5.3.1 Questionnaires and interviews

Written surveys are a familiar tool, which works well as a complement to expert reviews and user testing. The number of users participating in surveys is often rather large compared to the other methods.

To have a successful survey one should have clearly defined goals and prepare for the analysis work in advance. Precise questions increase the chance of receiving a result more likely to be able to use for guidance. Finding out how the users like various aspects of the interface is best done by using surveys with satisfaction scales and free form comments. This leads to a more practical evaluation. Performing the survey online or by email reduces the work of distributing and collecting papers [32]. This may also ensure a higher answer rate.

Using questionnaires is a method that does not study the interface itself, but collects the users' opinions of the interface based on a set of predefined questions. Interviews work similarly to the questionnaires but have greater flexibility and direct contact between the user and the interviewer. Questionnaires can be too time-consuming in proportion to the received information. However, it is seldom needed to have very statistically large volumes to be able to retrieve answers from the survey that can be considered to be reliable [10].

Before conducting the survey an assumption or a hypothesis should be formulated. In other words, the purpose should be clear before start. There should be something to confirm or reject [18]. Based on the results from the heuristic evaluation presented later on, the hypothesis became that new users would have trouble learning the system, especially where to find what they want, *and* that once learned the window management would slow down the work process and making tasks less visible.

When testing an existing version of a system to make improvements, users who are used to the system are of interest. Characteristics for these users are that they have thorough knowledge of the system at hand, put great importance in efficiency and few error situations, and have developed techniques for problem solving [18].

Including the users is also a good idea, since their resistance to change might be reduced if they can anticipate it [32]. Clients and support personnel are good sources for pointing out experienced users.

The users should be asked about typical tasks, message experiences (error messages in particular), and help usage. Surveys can also be a source to creating user profiles, which is why questions about previous experience and knowledge can be included. Other sources are focus group sessions, interviews and usability tests [18]. The profiles usually contain two types of characteristics: those that all users will share and those that might make a difference among the users. When selecting participants, users with differing experiences are preferable. A reasonable grouping is described below. If users from every group are not available, group 2 is the least important, because if both group 1 and 3 experience the same problems, one can be fairly confident that group 2 is likely to have the same problem [8].

1. Novice 0-3 months
2. Intermediate more than 3 months, less than a year
3. Experienced more than a year

As pointed out by Barnum [1], using a scale with odd numbers provides a risk that the users will tend to avoid the edges and go for the middle alternatives. However, based on recommendations from others [18] scales with 5-7 steps was used in the survey.

Open questions are useful but time consuming to analyse and compare. A limited number of such questions, especially at the end of a section with closed questions, can however result in interesting responses [1]. Following the outlines of the quality of user interaction satisfaction survey (QUIS) presented by Shneiderman [32], the questionnaire was structured with both types of questions (see Appendix B). It was also divided into areas to make the interpretation of the results easier.

Before conducting the survey, the questions were used as basis for the interviews. This was done to get a chance to see whether or not further explanations or follow up questions were needed.

The interviews were conducted with two persons from the same company, but with different tasks and different usage of Content Studio. Each interview was conducted one on one, and provided an opportunity for them to show their problems instead of just talking about them. Before asking the selected users to participate in the online survey, a pilot test was performed on one user. The interviews had already provided valuable input on which of the questions that were relevant. Still, the test showed that one of the questions needed to be rephrased, but other than that everything seemed clear.

The participants were asked via e-mail to conduct the survey. A period of two weeks was set as the time limit for incoming answers. After one week a reminder was sent out, resulting in several more answers.

Surveys and interviews both provided valuable information, but sources within the company could also be used for customer feedback. Examples of this are the technical support and personnel used to train customers [1]. To be able to sit in

some training classes was a good experience to see what problems the users had, and what they did not understand.

The data received from the survey was examined to see if there were unanticipated usability problems, and if it was consistent with the hypotheses. It was also compared to the results from the heuristic evaluation and the observations. The results of the comparison are presented in chapter 7.

6 Usability Guidelines

“Different types of guidelines are needed to evaluate different types of interactive systems.”

Preece et al, Interaction Design: beyond human-computer interaction.

To be able to perform the heuristic evaluation, a list of principles or guidelines was needed. The rapid growth of web applications, such as Content Studio, has revealed a lack of effective guidelines for their design and implementation. These types of applications where software is delivered as a service over the web could be thought of as part web site and part software application. Many web usability guidelines work well when applied to traditional web sites, as do several existing GUI guidelines for software design. However, neither set of principles alone is completely applicable to web application interface design [40]. Thus, tailoring a set of heuristics for evaluating Content Studio’s interface was needed.

Usability principles such as Nielsen’s [22] are similar to Norman’s design principles [27], and sometimes overlapping them, but they are mostly used to evaluate systems instead of designing them. Design principles often arise from experience, knowledge and plain common sense. They are intended to guide the designers through the process and help them remember what to include in the interface [30]. However, a principle is just a goal. It does not say anything about *how* it should be fulfilled. Instead, guidelines for a system have been developed from the principles depending on the future users and environment. The principle “be consistent” can, for example, result in guidelines such as “be consistent in the way you make users leave every menu”. In other words, one principle can lead to many guidelines, which can differ for different types of systems and so on [8].

Design principles, heuristics and guidelines and international standards are often too general and do not give any specific guidance for the construction. Platform specific guidelines such as Microsoft Windows User Experience are instead specific on a technical level, but too general for the particular domain. They are intended to be used as guidance for a designer, creating similar design of every information field regardless of which type of application that is build. Domain or product specific style guides are aimed at specific applications and based on the previously mentioned type of style guides. These are seen as a more specific design help and can cover a domain, a product family or even a single product [10].

In his alert box “The difference between web design and GUI design” [23], Nielsen clearly points out the differences between the two and emphasizes that different guidelines are needed when designing websites compared to traditional GUI design. This makes it equally obvious that different guidelines now are needed when designing a system that is a combination between the two.

Using existing guidelines for the web could affect the usability of web applications by restricting the interactions to those provided by the browser. The guidelines are therefore not flexible enough and are often not appropriate because the motivation of a web application user differs from that of a web site user. Overall, web-based applications differ from web sites in many ways, not just by being more interactive but also by being more likely to be used more intensively and more frequently [40]. And, according to Norman [28] as cited by Wroblewski [40], because users want the service that applications have to offer they are willing to invest more time in learning the functionality for the payoff of increased productivity. This behaviour differs very much from surfing of Web pages, in which case users will move on if they cannot quickly understand the site [25].

At the same time, interface design guidelines for software products do not address the behavioural patterns web users might have, and the limitations and possibilities of the Web environment. Still, traditional GUI guidelines are more applicable to web applications according to Nielsen [11]. These are aimed directly at applications not needing features such as “back”, “home” and so forth, which apply to many web applications. When designing an application to run in a browser instead of a desktop these constraints still need to be considered. Because of the Internet access, users might transfer both applicable and inapplicable knowledge if perceiving the use as similar to a web page. Interface guidelines must therefore be supplemented with additional guidelines from the web environment to fully cover the needs of interface design for web applications [40].

However, guidelines are very general, which is why they cannot be followed like a recipe expecting it to guarantee a perfect design. They will make it less likely to find severe usability problems when conducting usability tests, but have to be considered for the particular application in question and adjusted accordingly. Preferably they also need to be combined with other factors such as evaluation and user testing [8, 15, 40].

6.1 Principles and guidelines for Content Studio

Nielsen’s usability heuristics [22], Shneiderman’s golden rules [32] and Norman’s design principles [27], all have several items in common. For instance, they all emphasise the need for striving for consistency, putting the user in control and supporting the users’ limited memory. Windows official guidelines for user interface design [15] also contain several of these items, as do web specific ones. Being that Content Studio tries to resemble some of the aspects from the Office family, the principles on which Windows has based their design, should also be considered when evaluating the system.

Based on the most frequent principles, including some additional guidelines developed for web based applications [40], a set of usability principles have been put together to capture the most prominent ideas and at the same time better fit the interface design process for Content Studio. This way not applicable guidelines aimed for the web, such as ones regarding page layout and text links, does not have to be taken into consideration during further development. The guidelines can instead be complemented later with any project or company specific guidelines that might occur.

6.1.1 Consistency

Users should not have to wonder whether different words, situations, or actions mean the same thing [22]. A consistent layout across browsers and platforms will enhance usability within applications by maintaining a stable interface that users can rely on. Consistent aesthetics unite various sections of a web application and give it a coherent look and feel, but placing too much emphasis on aesthetics can cause usability to suffer. It is important not to allow visual treatments to overwhelm interaction elements [40].

Consistency makes the interface familiar and predictable, which makes the user more able to transfer existing knowledge to new tasks, thus learning new things more quickly. It is therefore important in every aspect, from names of commands to operational behaviour and placement of elements. This includes presenting common functions using a similar set of commands and interfaces and maintaining a high level of consistency for the conventions provided by the operating environment. The software benefits from the users' ability to apply interactive skills they have already learned elsewhere [15].

Inconsistent interfaces are the ones that allow exceptions to a rule. This is often arbitrary and makes it difficult for the user to remember. Consistent interfaces are therefore easier to use and learn. If a design is about to wind up with an arbitrary design aspect, the solution might be to standardize. Standards only have to be learned once, but in some cases the users have to be trained to learn them [27]. Consistent layout and terminology should be used through the whole design. Although, the existence of several forms of consistency makes this a hard rule to follow and a frequently broken one [32].

Use the conventions of link selection in web.

Web users are used to exploration and will move their mouse until a clickable area appears. The users' willingness to explore can make different forms of interaction possible. Both text and images can initiate actions and web users have come to expect this from their interfaces. The important thing is to make sure functions are not imbedded within hard to find or irrelevant elements [40].

Use common functionality for web applications and web browsers

Using interactions common to web browsers allows transfer of knowledge between web applications. As do standardization. Web users are familiar with the functionality of drop-down menus and checkboxes, they are easy to implement, and help conserve screen space. Exploit the similarities in the basic functionality of all web applications [40].

Avoid double clicks

Web users are accustomed to a single click when interacting with web sites. Eliminating double clicks therefore follows the established conventions of the web. According to Raskin [31] as cited in Wroblewski [40], double-clicking requires users to remember which elements that responds to a double click and what the outcome of the double click will be as opposed to a single click. At the same time, one must be aware that elements in a web application may resemble elements in client applications, which normally requires double-click.

6.1.2 Minimize users' memory load

The user's memory load can be minimized by making objects, actions, and options visible. Whether they are dragging an object or navigating to a location in a document, users should see how their actions affect the objects on the screen. The user should not have to remember information from one part of the dialogue to another. If not visible, 5-9 chunks of information is the average amount a user can keep in memory at once. Hence, instructions for use of the system should be visible or easily retrievable whenever appropriate. Visibility is also important, because users tend to recognize commands and options rather than recalling them [32]. Users remember a meaning associated with a familiar object more easily than they remember the name of a particular command. Using metaphors also allow users to transfer their knowledge, providing a direct and intuitive interface for user tasks [15]. Besides paying attention to the users' limits when it comes to memory, the designer must try to improve the feedback given by the system and let the user keep control [27].

Minimize the use of windows

A minimum number of windows reduces the mental load of managing multiple windows and reduces the possibility for windows getting lost behind others [40].

Use screen assistance

Rollovers and ALT-overs are commonly used on the web and users are familiar with them. Tool tips can clarify the use or function of interface elements and are present within many client applications. They also reduce screen clutter by revealing alternate choices or presenting additional information as it is needed. Although, too many rollovers may result in a flickering effect by constantly having visual items appear and disappear. Rollovers are beneficial for reducing screen clutter while supplying the necessary information when it is needed [36]. This makes rollovers particularly suitable for web application design. It should however not be counted on as the only clarifier for interface elements.

6.1.3 Feedback

Through appropriate and informative feedback within reasonable time, the system should always keep users informed about what is going on [22]. Depending on the type of action performed, the feedback can be brief or substantial, but every action should provide some sort of feedback. The user should even be informed when a set of actions is complete, so they can relax and drop the task [32]. The quicker the response is, that an action has been performed and what the outcome was, the more smoothly the interaction becomes [27].

Microsoft [15] distinguishes between good and effective feedback. The first confirms that the software responds to input and helps distinguish the nature of the action. The latter is presented as close to the point of the user's interaction as possible. Even when the computer is processing a particular task, one should provide the user with information about the state of the process and how to cancel the process if that is an option.

Use underlined fonts as hot-spots

Actions can be embedded within text-based explanations and allows execution as part of a natural sequence.

Use motion cues as a feedback mechanism.

Motion cues can be used to show continuity and dimensionality in transitions, illustrate change over time, enrich graphical representations or attract attention [25]. Poorly implemented animation may on the other hand distract users. The interface should not depend on motion cues for understanding.

6.1.4 User control

The user should always feel in control of the software instead of feeling controlled by the software. For instance, one should assume that the user - not the computer or software - initiates actions and plays an active rather than reactive role. If tasks are automated, the user should be allowed to choose or control it. Because of the varying skills and preferences among users, they must be able to personalize aspects of the interface. The software should reflect user settings for different system properties, such as fonts and colours and be as interactive and responsive as possible [15].

Users often choose functions by mistake and will need a clearly marked way to leave the unwanted state. Users like to explore an interface and often learn by trial and error. Thus, an effective interface supports discovery at the same time as it prevents errors, which will be further discussed below.

Make actions reversible or recoverable

By permitting easy reversal of actions and supporting undo and redo, the system helps the users relax and encourages exploration, while still feeling that their actions result in the interface responding to them, not the other way around [22, 32]. At least, the system should provide appropriate sets of choices and warn users about potential situations where they could damage the system or data.

6.1.5 Prevent errors

A part of designing explorable systems is to help users recognize, diagnose, and recover from errors. Foremost, one should design the system to prevent the user from making serious errors. However, if an error is made, ways to recovery should be supplied by the system. One should always assume that an error that *can* be made *will* be made, and therefore plan for it. Or, as also stated by Norman [27]: “make it easy to reverse operations, make it hard to make irreversible actions”.

Error messages should be expressed in plain language, indicate the problem, and suggest a solution, but a design which prevents a problem from occurring in the first place is even better. Either situations that are likely to result in errors should be eliminated or the users should be presented with a confirmation option before they commit to the action [22]. Even in the best-designed interface, users can make mistakes. These mistakes can be both physical such as accidentally pointing to the wrong command, and psychological such as making a wrong decision about which command to select [15].

Make unwanted actions unavailable

Unwanted actions should not be possible to accidentally perform [30]. At any certain point, the user needs to be restricted or constrained from unavailable actions. For example, unavailable menu options should be deactivated. Another type of constraint is called forcing functions, which forces the user to perform one action before engaging in another one. For instance, the user might have to specify a required setting of a component before changing another. This also eliminates the use of an error message reminding the user to do so. There is then only one possible thing to do – the right thing [27].

6.1.6 Flexibility and efficiency of use

The system should be able to cater to both inexperienced and experienced users by trying to add features supporting them. Allowing the users to tailor frequent actions and providing shortcuts may for example often speed up the interaction for the expert users [22]. Users learn better if the knowledge required for a task is available externally. The design should not hold back actions especially for the experienced users who have internalized the knowledge. It should be easy to switch back and forth [27]. Keyboard shortcuts eliminate the need to reach for the mouse, find an option and select it, favours memorability and make transition from novice to expert easier. Toolbars, icons and palettes all supply the user with actions to apply on the specific object, available with a single click [32].

To motivate the user to learn more about the systems they have to be rewarded in some way. Shortcuts make it possible to work more efficiently and are a good reward for learning about the system. This in turn can inspire the user to learn even more. Furthermore shortcuts make it possible for the user to use the keyboard to a greater extent and thus lessening the risk for mouse related injuries. Keyboard shortcuts, function keys and toolbars are all examples of different types of shortcuts [29].

The challenge in designing for advanced users is providing efficiency without introducing complexity for less-experienced users. Developing shortcuts is often a useful way to support these users. In addition, advanced users may be dependent upon particular interfaces, making it difficult for them to adapt to significant rearrangement of or changes in an interface [15]. Another way to support different users is to enable them to personalize the environment [30].

Allow for customization

If the users are allowed to customize the interface by changing some settings or arranging other objects as they wish, the efficiency can be further increased.

6.1.7 Simplicity

Tasks should be simple in structure, minimizing the amount of planning or problem solving they require [27]. Keeping the display simple, and allowing access to online information helps the user when performing a task. Simplicity should not mean lack of functionality; it should mean fast initial learning curve and consideration for the number of concepts the user has to understand [3]. Making something simple to use often requires a great deal of work. Maximizing functionality and maintaining simplicity often work against each other in the interface. An effective design balances

these objectives. Simplicity also correlates with familiarity; since things that are familiar often seem simpler. Whenever possible, one should try to build connections that draw on the users' existing knowledge and experiences. By hiding inappropriate information from the users the amount of information the user must process becomes reduced [15]. This leads to another aspect, namely visibility.

Make information visible

Every extra unit of information in a dialogue competes with the relevant units of information and makes them less visible. Dialogues should not contain information which is irrelevant or rarely needed [22]. The more visible functions are the more likely the user will know how to use them and what to do next. Visual attributes provide valuable impressions and communicate important cues to the interactive behaviour of particular objects. A coherent environment that clearly contributes to the user's understanding of the presented information should be provided [15].

There are some laws that manage our vision. These come from the Gestalt theory, which is a way to visually connect information, by using the laws of proximity, closure, similarity and good continuation. These show how similar objects or objects close to each other are perceived to be connected and can thereby be seen as a group.

Providing actions that match interpretations and making outcomes of a system obvious makes it clear what is possible to do. The term mapping describes how the relationship between controls and their effects is shown. Clear mapping gives the user better understanding of the effect of an action [27]. System that provides affordance allows users to know how different aspects of it should be used. I.e., it should be obvious what can be done to objects in an interface, but according to Norman [27] screen-based interfaces has learned conventions and thus perceived affordances, which is why other design principles, such as feedback, can be more useful.

Open the browser window to full screen size.

Full screen maximizes the display, allows the web application to be in focus and hides the navigation tools of the web browser. The elimination of the web browser tools, borders, and menus makes the browser less confusing. Thus, allowing users to utilize the interaction models of the web application instead of the browser. A possible problem is that inexperienced users may not expect such actions from a browser. If it is possible the use of animation could help clarifying the browser's actions [40].

Speak the users' language

When working for simplicity, it helps if the language follows the same pattern. The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order [22].

6.1.8 Help and documentation

Any information for help and documentation should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large [22]. Certain systems can take longer to learn, but help such as tutorials and manuals should always be available [30]. However, to supplement ease of use and support efficient

and complex operations, one will inevitably need to provide some form of user assistance.

Provide contextual help

The ideal form of user assistance integrates seamlessly into the overall design of the application. Its interface should be simple, efficient, and relevant; the interface should enable the user to get help easily and then return to a task [15]. Contextual help does this. Examples are tool tips, and help areas connected to specific objects.

6.2 Trade-offs

When designing a system, trade-offs can arise when applying design principles. For example, constraining an interface can make it less visible, and making it easy for novice users can make it too simple and unrewarding for expert users. Consistency is another principle that is likely to cause problems. Following one convention can make it inconsistent with another one. Consistency or any other principle often helps, but many situations require trade-offs which forces the designer to focus more on another aspect of the interface. Trade-offs are always present, and it is up to the designer to decide which rule that is most important to follow. However, the principles provide good assistance and designers can use them to make better designs [2].

7 Results

The survey confirmed that most of the users work as marketing assistants, web editors and administrators, web and system developers, consultants, IT technicians and IT managers. Their use of the system differed depending on their work tasks. Overall, they mostly used Content Studio to edit and update information such as text and images, to make changes in the existing templates or build their own, and to create functions and Active Scripting components. Nearly everyone that participated in the survey had knowledge in layout programs such as Photoshop and Illustrator. About 40% had prior knowledge of HTML either through programming or editing programs such as Dreamweaver and FrontPage. Everyone had knowledge of the Office family, in particular Word processing, and some had extensive knowledge in development programming and network administration. Over 60% of the users had never used a similar system before. Those who had, had either used a self developed one or EPiServer, SiteManager or Mambo.

7.1 Expert input

The result of the heuristic evaluation showed that many of the guidelines were broken, although some more than others and some not at all. After performing the heuristic evaluation a few more concerns arose apart from some more specific details that were discovered.

The first concern was if new users (who have not seen this particular program before) were able to become productive quickly? A concern also rose about how users will be able to work with more advanced features after they have had the product for some time. Yet, basic tasks were considered to be a bigger concern, since if users are unsuccessful when introduced to the system they might not continue to use it. Whether or not new users would be able to navigate through menus quickly and easily to find what they want was also a question of interest. As were if all functions and icons, together cluttering the workspace, really was needed. Another thing that cluttered the workspace was the amount of open windows, which might be a concern if they slow down the productivity.

Content Studio fulfils the usability goals to various degrees. It was found to be very effective, safe and memorable, but might need reinforcements to become more efficient, learnable and to achieve a higher utility.

The guidelines were looked at one at the time, while examining the application, and the problems found are completely listed in Appendix A, but the most important ones are listed below:

Consistency

The back-button in the web browser toolbar can sometimes work as if it is a part of the program, but it does not belong to it since a back and forward button also exist in the web view. This affects Content Studio because if the user selects a folder and clicks on back, the application needs to be restarted. Unless he or she has used the web view and navigated to at least one link, then nothing happens. Because of the web browser there also exist double sets of "file", "view" and "help", which might cause confusion (see figure 7.2). Some of the checkboxes in different dialogues cannot be checked even though they are white and, thus, indicating that they are active compared to the grey ones (see figure 7.1).

Secondary buttons (right mouse button) are typically used to copy already available functions for primary buttons (left mouse button) and thereby providing a shortcut for the user. In Content Studio, many functions exist only in the popup menu.

Icons in the dialog for active scripting components are placed in a different order compared to the ordinary view. The same buttons in dialog boxes for units and categories are also placed in a random order. This applies to some other options they have in common as well. The amount of buttons has also resulted in two rows of buttons on top of each other, which is difficult to get a complete overview of (see figure 7.1).

Minimize users' memory load

Single-click and double-click are sometimes following web standards and sometimes software standards. The mouse interaction is further confused through difficulties remembering what options the right mouse button contains.

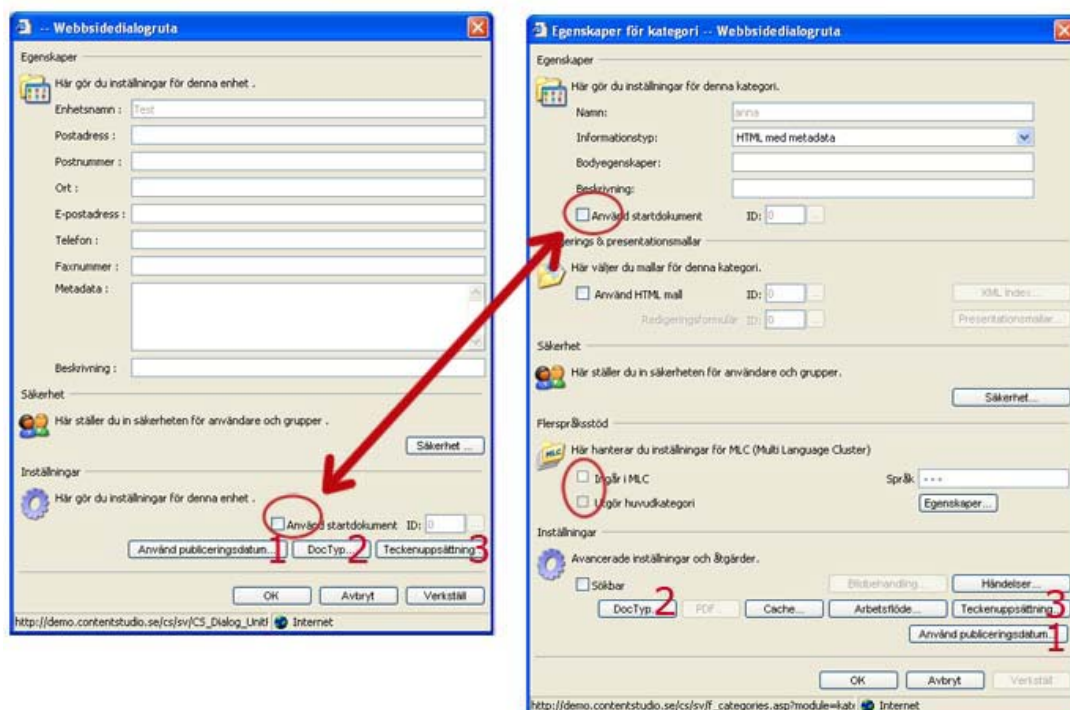


Figure 7.1. Inconsistency for placement of buttons and options in similar dialog boxes. The vertical circle to the right illustrates non selectable checkboxes, where one is white and the other grey.

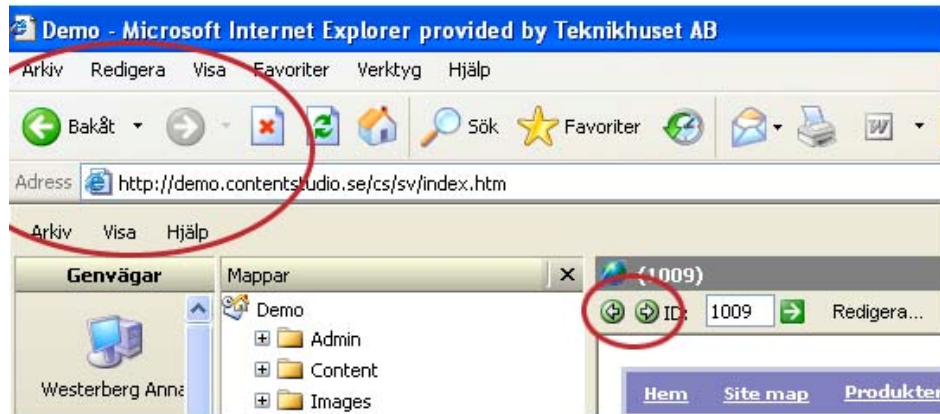


Figure 7.2. Double sets of menus and navigation buttons.

The use of popup menus, which appear on the location of the mouse, reduces the clutter in the UI and is only showed when the users request them, which is good. But at the same time, their length must be held to a minimum and they need to be as small as possible, since they cover the display. Several of the dialog boxes for active scripting components are very long and sometimes ending a section just before the window edge, which can cause users to believe that no more options are available.

Feedback

When leaving the application, exit does not give any feedback of the fact by asking for conformation. Closing the web browser with an opened document inside will result in any changes being lost.

There are text areas that are clickable but there is no indication given to show this fact. For "insert image" or "open" a preview option is available, but no indication that the text is selectable is given. Instead, preview can easily be mistaken for the already open frame, which is very small and sometimes showing objects that are very big. Other problems with proper indications were found in several items. For instance, the icon "check out" should make anything not selectable unavailable to more clearly indicate the need to click it (see Figure 7.3).

User control

Even though the menu items are greyed out, the icons next to the item text is not. If an item in a menu is unavailable, the entire item and not just the text should be grey. An undo button is always available, making small changes easy to reverse and allowing the user to feel more in control.

Prevent errors

Content Studio is equipped with revision control, which creates a new version of a document every time it has been checked out from the repository. This supports exploration in an excellent way, since the user always can retrieve an old version and therefore does not have to worry about making a mistake that cannot be undone. However, errors do occur and as can be seen in figure 7.4, they can be more informative for the user.

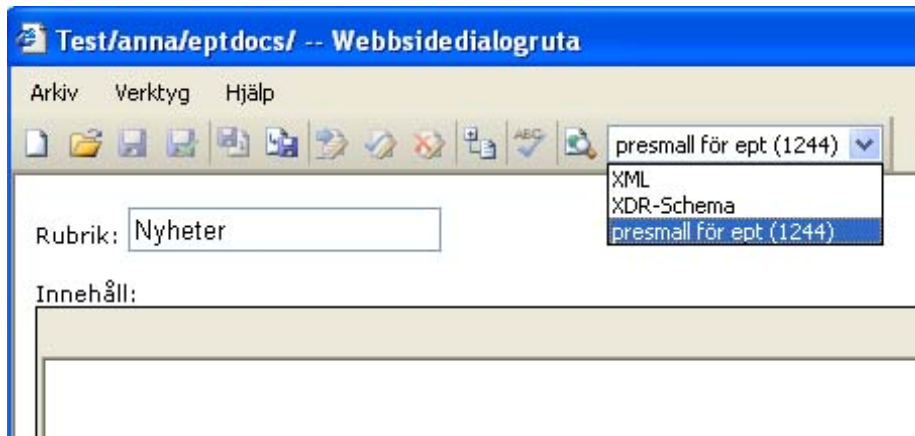


Figure 7.3. To be able to edit this document, one has to select check out (6th icon from the left), but there is not enough information constraining the user to perform this action. For instance, all three of the drop down menus are expandable, and not greyed out.

Flexibility and efficiency of use

When it comes to supporting both experienced and inexperienced users, Content Studio is good at providing icons and other shortcuts. Especially the simple mode helps novice users to edit information in a simple way. However, not every shortcut is available as a menu option and may only be accessed by a right click or an icon, for instance.

Simplicity

Content Studio provides information in an order that should correspond to the users' priorities when starting the application. First shortcuts are presented, then a folder overview to enable the user to quickly navigate to a desired document. This far it has a very simple design, which makes the information visible, but when one gets more used to the application some problems, such as error messages, become visible instead. A message such as the one in figure 7.5 does not provide enough information for the user to make an informed choice. When it comes to language, messages such as these are too technical and not at all user focused. This applies to other areas as well, for instance “_blank” is used as explanation to select a link to be opened in a new window. This is difficult to understand and remember for inexperienced users.

To look at a page in the web view, one can only drag it to the icon. Just selecting the document and then clicking on the web view does not work. The drag-and-drop function is not obvious and to drop it on the correct spot, one has to cross the web tree where the pointer might show that it is not allowed to drop objects or, if touching the objects in the area, the tree will expand and allow the document to be inserted. This means that accidentally dropping the document will copy it into the



Figure 7.4. Error message.

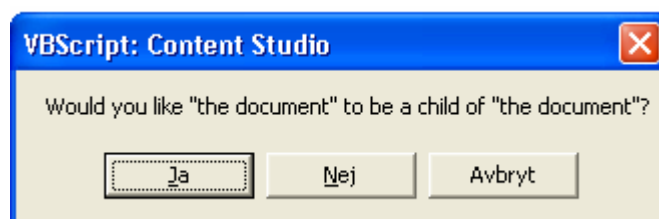


Figure 7.5. Question displayed when adding a new item to a menu.

content structure. That the drag-and-drop is supported for inserting text from other documents or pages (both inside and outside the application) is very good and a quick way to copy information.

When using a small screen, the workspace gets very cluttered. Windows on top of each other makes it difficult to locate the ones at the bottom. Especially when the top ones are dragged around to find them, in turn making them hard to find a minute later (see Figure 7.6). They also cover the document information and web tree, which is needed to view other documents.

Help and documentation

When requesting help, the user is forced to log in instead of being linked directly to the area in question. The help does not clearly provide a solution to the problem and even though the user has actively requested help, the support area is not shown. Instead, the user is forced to navigate to it.

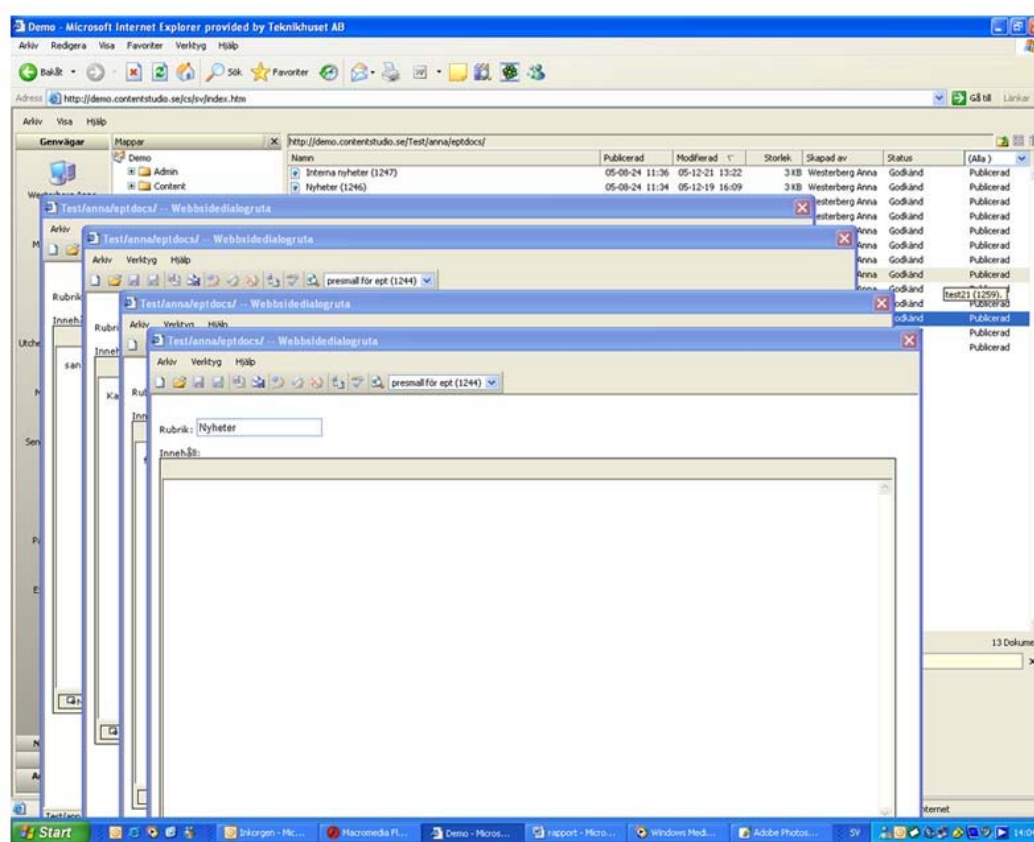


Figure 7.6. When opening documents for viewing and editing, they all are placed on top of each other.

7.2 User input

7.2.1 Interviews

Two formal interviews were performed. The first one regarded a system architect, A, who spent 5-10 hours a week on the system and used it mostly for programming and development. A's general impression of Content Studio was that it was neither difficult nor easy to use, but at the same time that it would be difficult to make it easier without sacrificing functionality. The amount of information displayed on the screen was considered to be sufficient, but might be cluttered for smaller screens. Because A spent much time programming, it was desirable for information such as function calls to be more similar. Names of functions were said to not always describe what they actually do and can be inconsistent. Computer terminology had been noticed to be used a lot but A did not find it disturbing. Although, A believes that it could affect many people into not knowing how to use some functions.

A did not perceive the interface to be supportive when not listing built-in functions, which forced A to keep much information in the head. When new functions are released A wanted a newsletter to be sent out.

Finally, A never used the online-help because it took too long. The biggest benefit of Content Studio was considered to be that it was quick to build in, that the written code was reusable, and that it was easy to integrate. Its biggest drawback was the lack of documentation.

The second interview was with a person in charge of a web department, B.

Content Studio was always open at B's computer, making it possible to make small changes now and then throughout the day. Editing text and graphics were the most frequently used functions together with the HTML view to manage layouts. B found the interface to be very familiar because it looked so much like a Microsoft application, but overall, B found the design to be inferior. The language included many technical terms, but B knew HTML before using Content Studio, which simplified some parts. At the same time B remembers to have helped others not as experienced in the programming language, which indicates that the language might have been too technical for some. Furthermore, the language was inconsistent in the way some dialogues were in English and some in Swedish. B did not always understand how to fill in a dialogue and sometimes called for support, but assumed that the dialogs were meant to be self explaining. Supposedly the support would want fewer calls of this type. B had also experienced problems when discovering that it was not possible to copy text from one page, close it and then paste it onto another open page. One had to copy and directly paste the selected text into the other document. If the original page is closed first, the option to paste became unavailable. Because this was a rather frequently performed action for B, forgetting to leave the document open caused much delay during a day.

B only used the web tree to create new documents, otherwise the web view was used to navigate to the right page. Exploring the system felt easy and safe. In the beginning B had wanted easy access to an online-help, but had not used it in later versions of the application. B had not even noticed the online help option in the top menu and did not know how to log in on the support site. Information about updates was mentioned as a current need of help.

B claimed not to use the menus much, but at the same time it was observed that B always used the file menu to check out documents during the interview. Furthermore this was often forgotten and B tried to edit the text without selecting check out.

B liked Content Studio because it allowed the company to develop a lot with the competence they already have in the building, and that it was not just a tool for editing. B emphasises that one must realise this before purchasing it. “You don’t want to buy a Porsche just to listen to the radio”. The biggest downside with Content Studio, according to B, was the messy interface.

7.2.2 Observations

During informal interviews and observations some additional problems was discovered and some concerns was reinforced. For instance, a misleading mapping and lack of feedback were pointed out. When clicking on a category in the tree structure to the left, all documents in that category is shown to the right. If one thereafter selects the icon “my documents”, all documents are shown, but the previously selected category is still highlighted. This makes it look like the documents showing to the right belong to that category. The highlighting should of course follow the latest selected object. When viewing “my documents” some users wanted a column for document type, or maybe even category. This requires a possibility to change the column widths, which was desirable anyway.

Observing the users trying to perform different tasks, showed that the dialog boxes for properties were difficult to remember how to use. In particular it was difficult to remember where to find various settings. Remembering to check out the documents before use was also difficult and seemed to frustrate the users. The dialog boxes for active scripting components, for instance, are very long and made it difficult for the users to find wanted options, thus reinforcing the result from the heuristic evaluation.

Another thing that became very obvious was that Content Studio encouraged exploration. None of the users were worried about making mistakes, and gladly tried different options. However, some of the messages they received when poking around in the interface was too requesting instead of encouraging, which some of the users reacted to.

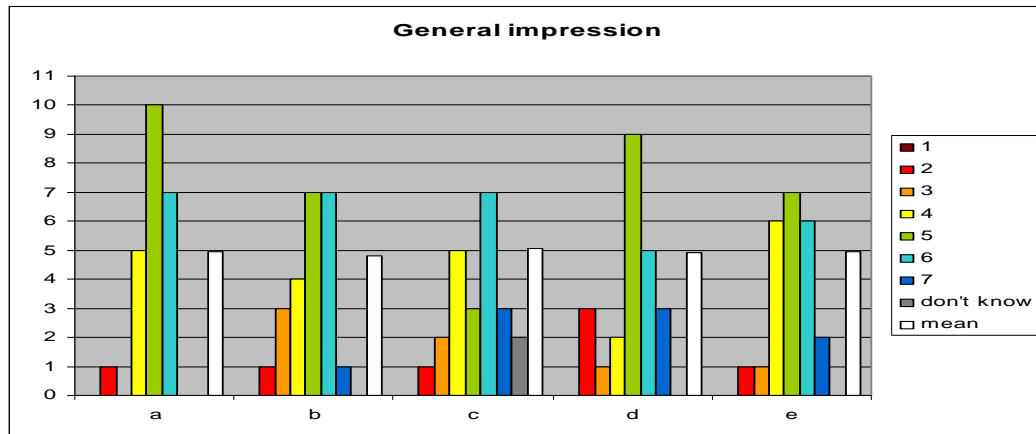
7.2.3 Survey

The result from the questionnaires was gathered. The answers from the questions corresponding to a particular area were put together in a diagram, displaying the number of participants for every option, and the mean. The options ranged from 1 to 7 are represented from left to right in ascending order, followed by any answers that were indecisive and the mean. The questions in full can be viewed in Appendix B. The free form comments are summarized previously to the diagram. An asterisk behind the question means that someone had neglected to answer that particular question.

General Impression

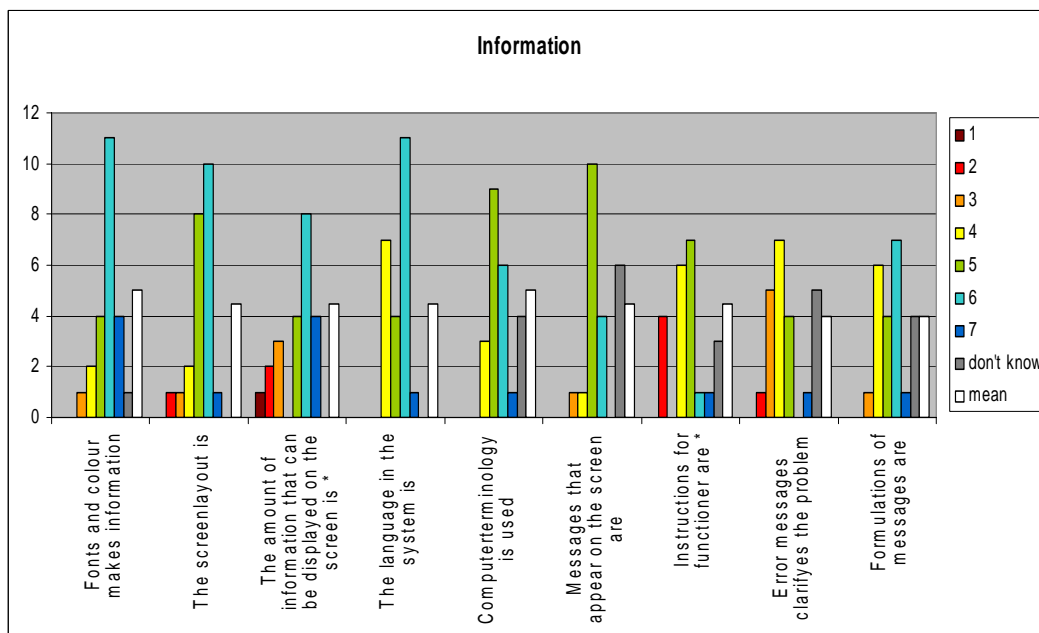
The general impression seemed to be that Content Studio is a very capable program, and the difficulties lay in the user’s experience and knowledge, although many things

are possible to accomplish without any programming skills. Overall, the impression was positive, but many of the participants acknowledged the fact that time is a factor when it comes to learning. However, the novice users seemed to be under the impression that anything is possible when a certain level of knowledge is reached and seemed to be willing to trade a learning period for it. Hence, the threshold for being able to use Content Studio for development appears to be quite high, while usage for administration is much simpler. The things that were mentioned as downsides, and thereby holding down the general impression was: lack of documentation, some difficulties with the AS components and a somewhat frustrating interface.



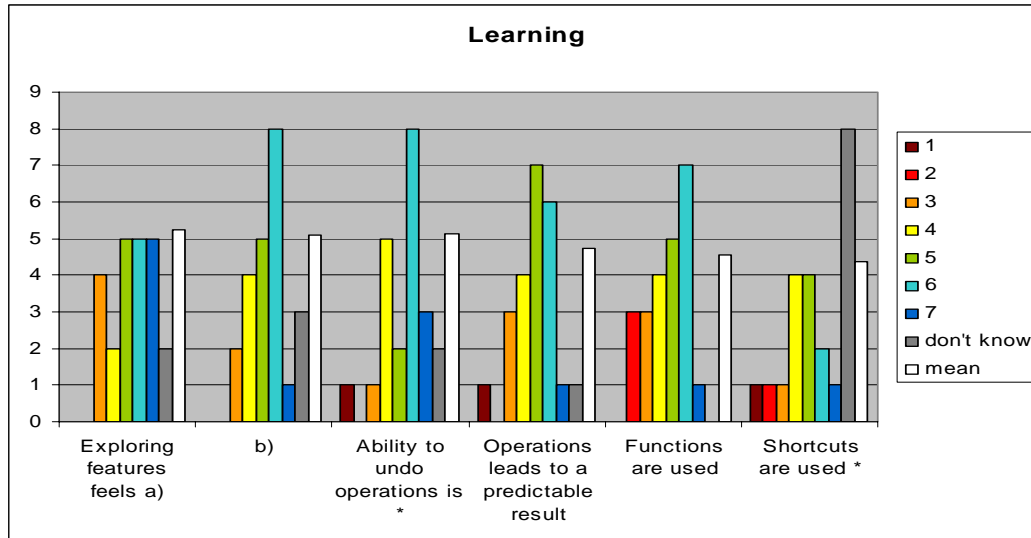
Information

The feedback on how well information is presented and viewed was mostly focused on insufficient descriptions of components. A wish that the view of lists of documents would be more flexible was expressed along with a possibility for colour coding. The window management could be more efficiently arranged, to enable a possibility to switch between open documents. Other things mentioned were a too technical language for non technicians, such as editors, and non declarative messages for script errors.



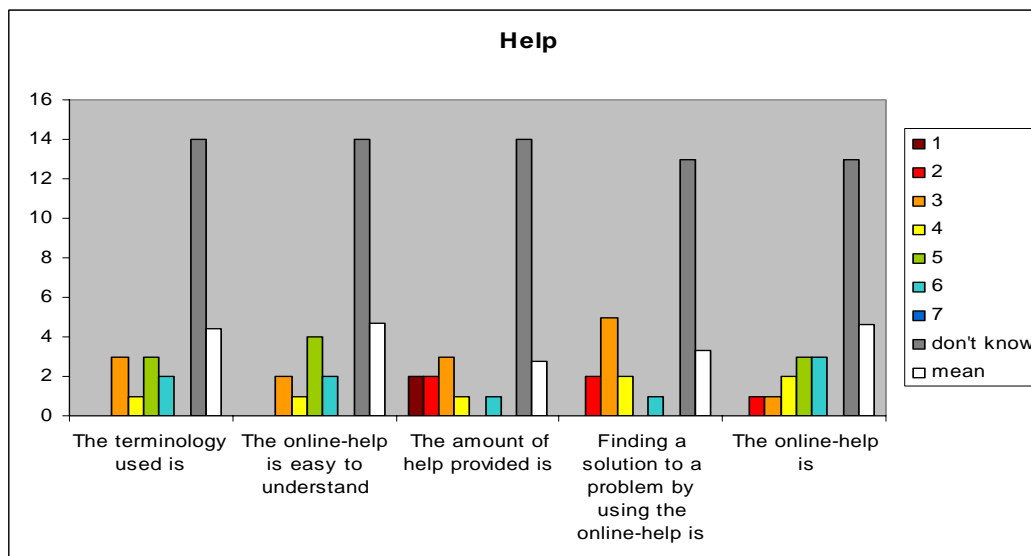
Learning

The few comments that were made were about training and requested more after having worked with the system for a while.



Help

Half of the participating users had never used the help or had used it previously but found it to be insufficient, and therefore never used it again. The users that had used the handbook understood it better after an explanation was given. Reasons for not using the available help were difficulties finding technical information and sometimes uncertainty regarding whether or not they have access at all. More tool tips were requested by several users.



Additional comments

The question on what they like most with Content Studio resulted in a general answer that highlighted flexibility, powerfulness, and ease of use when it comes to making changes. Several users connected this to the WYSIWYG mode and the similarities Content Studio has with Word. The AS component system also received praise from the users, due to its ability to make it easier for them to combine them into powerful functions and the possibility to create their own components. The customers seemed to appreciate that Content Studio puts the control in their hands.

On the question to what they did not like about Content Studio many of the participants commented on the inadequate function descriptions and lack of documentation. Although liking the tutorials, they were considered to be too narrow and too few. The users also had problem knowing when new releases or updates is made available. Again, the wish for colour coding was mentioned along with a request to be able to structure the code and be able to keep the changes. The size of windows and the lack of flexibility of the interface, such as column widths and moving documents in the folder hierarchy, were also disliked. This sometimes could be connected to a feeling of limitation and that some of the information felt locked away. The menu system was found to be messy by some, who also complained about the file management and wanted more drag and drop functionality. Other downsides were incapability for the user to create tables and images in Word and use them in Content Studio, and accidentally being able to close the browser window which forces the need for a restart.

77% of the participants were certain that they would recommend Content Studio to others. Reasons for doing this were the diversity and flexibility of the program; that it was considered fairly easy to work with despite their lack of HTML knowledge; and the excellent support Teknikhuset provides. Other comments were requests for a better user manual, more tutorials and a support to call that is always present. One made a comment on that he or she already had told Teknikhuset about his or hers problem. This knowledge brought some additional comments to the surface, after finding the wish list of customer input the support previously had put together. Suspecting the results to depend on the users' experience, new diagrams for the areas were put together. These showed in several cases that the satisfaction increased with the amount of experience. In other words, expert users rated aspects higher than novice users. The diagrams can be viewed in Appendix C.

7.3 Research from other systems

Investigating the competitors' strengths and weaknesses and comparing them to an existing version of a product can help designing the product to fulfilling the users' needs in a better way than the other systems have [8]. Also Molich [18] encourages that one tries to learn from others by looking at other sites and learn from them.

Future systems, such as the new versions of Windows and Office, can have a big impact on Content Studio if it becomes popular. Microsoft has done some major changes and if Content Studio wishes to continue to follow their lead, they might have to adapt to certain ideas.

7.3.1 EPiServer and SiteVision

Comparing the usability of competing systems is always profitable. However, not being able to experience the application first hand, the review was limited to screenshots from an online demonstration.

The first impression of EPiServer was that they had put a lot of effort into making it look easy to use. It seemed more focused on the novice user and less technical than Content Studio and had more comprehensible words on buttons, tabs and menus. Overall, it had more declarative texts, such as “open in new window” compared to Content Studio’s “_blank” to tell the system to open a link in a new window. The dialog box for link properties contained a possibility to decide if the link should be made up by text or an image and so on.

EPiServer had a logotype that separated the application from the web toolbar, which increased the feeling of a separate program. It also had CSS (Cascading Style Sheets) templates for images. Together with their templates for forms, image galleries and calendars this helps the user create results. This way the user receives help to reach his or her goals without having to possess the knowledge of how. The application used tabs for switching between different views, which is a common way to not having to deal with many open windows at the same time.

On the downside it was not obvious how to switch between edit and other modes and EPiServer used tabs almost everywhere, which could make it difficult to get a simple overview. For instance, they used tabs inside other tabbed environments, which could cause confusion for the users.

SiteVision provided tree structures for both content and navigation, which is recommendable, but not by displaying one under the other. This makes the structure messy and creates problem for very long tree structures. SiteVision displayed pages in a way that first opened a page inside a frame in the window, but then supplied a small popup window for editing the same page. It seems as if only one page at the time can be opened and viewed and the popup solution for editing is not preferable. SiteVision had placed a lot of emphasis on templates, or modules as they liked to call it, which further strengthens the notion that templates are a good way to go.

The accessible material was substantially less than for EPiServer, which is why further opinions would be mere speculations.

7.3.2 New versions of Windows and Office

Including an investigation of the new versions of Windows and Office in this work was requested by the assigner. This was due to the fact that Windows applications previously have influenced the appearance of Content Studio and their new look can affect the future Content Studio even more.

Since the new system comes with many new features and technologies, this review focuses on changes made to the graphical interface and the work flow. Not having access to a beta version reduced the information on which the review could be based. Instead it was limited to screen shots and other reviewers experience from live previews. According to Microsoft [16], the new Office UI was designed very specifically for Word, PowerPoint, Excel, and Access. Although it is not intended to be a model for all other Microsoft applications at the moment, other user interface designers may be able to apply some of the principles from the new design to their own design challenges.



Figure 7.7. A set of Command Tabs for the next release of Microsoft Office Word [16].

Because Content Studio relies so much on the familiarity of the icons and structures of Office, a future adaptation might have to be made to the new UI if it is accepted by the users.

Office 12

Previous versions of Microsoft Office use menus, toolbars, task panes, and dialog boxes as the way for people to interact with the application. However, adding new features that people easily could find and use, Microsoft admittedly says became more and more difficult. The primary design goal for the new UI has therefore been to allow users to more successfully find and use the more advanced features. Another goal is to preserve an uncluttered workspace and reduce distraction for users [16]. All and all, they like to call it a results-oriented approach that will help the users to better focus on their work. Commands are said to be better organized and presented in a way that corresponds to how people work.

The new UI contains a ribbon, which replaces drop-down menus, toolbars and dialog boxes. The ribbon is an area at the top of the screen that holds formatting tools for an array of Office options, and the tools change both on command and in context when the user works with different elements [14]. In figure 7.7 for example, items such as “Write” and “References” do not lead to drop-down menus. Instead the Ribbon changes, revealing different sets of tools. The result-oriented approach enables the user to often get WYSIWYG thumbnails of the results they will get if choosing an option. The whole goal of these changes, the company says, is to put features at the users’ fingertips which are currently buried in menus.

Early tests are said to have indicated that even though the look of the applications is new, people quickly become accustomed to the way they work. According to Microsoft [16] this is due to the simplicity of some of the new interface features such as command tabs, contextual command tabs, galleries and live preview (see figure 7.8 and figure 7.9). Command tabs have replaced the traditional menus and toolbars. They display the commands that are most relevant for each of the task areas. For example, Word has Command Tabs for writing, inserting, page layout, and so on. Excel has a similar set of Command Tabs that makes sense for spreadsheet work. Contextual Command Tabs contain sets of commands that are only relevant for specific objects. When the user clicks on a table a contextual command tab appears with commands that are relevant for table editing.

Because they only appear when they are needed, the commands needed for the current operation become easier to find and use (see fig 7.9). Galleries will provide the users with a set of results to choose from when working on, for example, a document (see Figure 7.8).

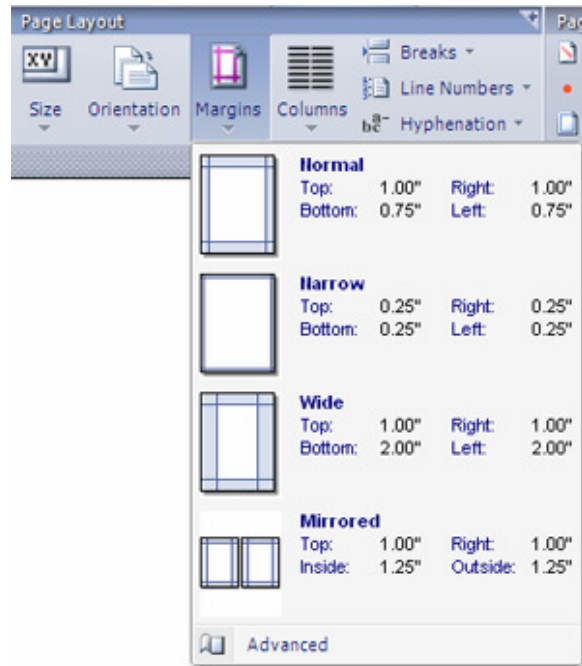


Figure 7.8. Galleries, here for margin layouts. The usual settings are available through the “advanced” option [16].

Instead of presenting a dialog box with several options, a set of potential results which simplifies the process is shown. For those who wish to have more control, the traditional dialog box interfaces will still be available. Live preview shows what the results of applying a formatting change will be as the user moves the cursor over the gallery. This assists the users and enables them to make choices and create results with less time and effort.

The whole interface seems to help the user see available choices without having to choose, which supports experimentation. Selecting a text makes a small toolbox with the most common operations visible. If a user is editing text far down in the document, the toolbox enables him or her to make corrections without having to move all the way up to the main toolbar to switch tab. This supports Fitt’s law, which is a predictive model of the time it takes to move a cursor and point to an object. It is well suited for user interfaces, since the time it takes to reach an object is depending on the distance and the size of the target. The basic idea is that the further a user has to move the mouse to get to an object, and the smaller the object is, the harder it is to locate and target [4].

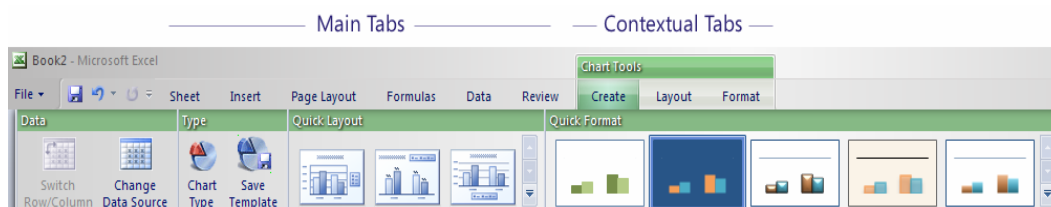


Figure 7.9. Command tabs, both contextual and ordinary [16].

Windows Vista

Microsoft describes Windows Vista as being confident, clear and connected. The redesign is made to be more intuitive, more attractive, and easier to use, thus making the work more efficient. The fact that the users are already familiar with the technology, Microsoft states, will result in minimal need of training and support.

Several rules and guidelines are set up for when designing a Windows Vista user experience [17]. At the time for this paper, the Windows Vista UX Guidelines downloaded from Microsoft download centre, were still a part of the preliminary documentation and might therefore be changed. Several of the chapters were not even written.

Many of the rules, besides the ones instructing developers how to make use of the new look, encourage developers to create well written and useful dialog boxes and messages, design for exploration and clean up the user interface. These are consistent with the ones previously described, and do not bring any revolutionary ideas to the table. Still, the ideas on how to design a great user experience [17] is profound and worth considering when designing an application. They too keep the designer thinking about providing a clean interface, creating good explanations to the solution of a problem and having the right set of features in the right places. Other ideas are to prevent errors and to provide help links for helpful and supplemental information, and of course to test the UI both before and after launch.

Microsoft Windows Vista introduces a UI with new features and enhancements for both end-users and developers. To ensure that one's programs will be consistent with common Windows interfaces and applications it is important to be aware of these changes.

Conclusions

In his alert box "R.I.P WYSIWYG" [26] Nielsen claims that we have reached the limits of the current GUI paradigm. Displaying commands in menus, toolbars, and dialog boxes works with a limited number of elements. Applications such as Microsoft Word 2003 have 1500 commands, and users typically have no clue where to find most of them. According to him, direct manipulation of WYSIWYG objects have some limitations including the amount of commands for the user to locate and no guidance on how to reach the goal from the starting state. Here, Nielsen also supports Microsoft's claim that people generally are better at modifying an existing design than creating one from scratch.

Until a beta version is released, it is hard to say how the new interface will be received by the users. Nielsen, along with every one else who has not worked with the software for an extended period, wants to hold off the final opinions. However, he does say that the new design does seem to resolve many of the problems with the current user interfaces. If the new interaction style works as well as early predictions indicate, users will quickly expect many other user experiences to provide a results-oriented design [26]. McCracken [14] believes that the reaction will be negative, based on Microsoft's history with Office interfaces. As an example he mentions the paperclip assistant, which was presented as a feature said to be loved by both novices and experts. He thinks there is a chance that the UI will work for both new users and the experienced ones, but raises the question on how the inconsistent look will affect the reception. The new Office is not even going to appear in all the Office applications. Users will have to get used to switching back and forth between applications with the old look and ones with the new look. According to McCracken, this will be the foremost reason for a quick influence of other applications as well or the reason the new interface will not work at all: "I don't see a scenario in which the

Microsoft apps behave completely different from every other program on the market”.

It will be very interesting to see how these new thoughts will apply to other environments. When it comes to Content Studio, several ideas are applicable and can be used regardless of the success of the other applications. The galleries are one example. Content Studio already uses components ready for use when providing Active Scripting, but can expand this feature into including other types of templates as well. The more result-oriented way of thinking can be a good way to further support the users when they are developing on their own.

7.4 Summary

According to Dumas and Redish [8] problems should be organised by scope and severity detecting how widespread and critical the problem is. The levels of severity are:

Level 1 – problems prevent completion of the task

Ex. Users consistently chooses the wrong menu options or gives up after a few tries.

Level 2 – problems create significant delay and frustration

Ex. Lack of feedback causes users to do the task again

Level 3 – problems have a minor effect on usability

Ex. Using the same word for different actions.

Level 4 – problems are more subtle and often point to an enhancement that can be added in the future. Ex. “it would be nice if...”

How severe a problem is, is often a combination between how often a problem occurs, how difficult it will be for the users to overcome and how persistent the problem is [22].

None of the methods used for evaluation found any problems that were categorized as level 1. The window management issues were classified as level 2 problems along with some feedback problems. The present help functions were classified as level 3 severity, because they mostly need to be more accessible. Many of the minor problems are things that work as they are, but which would work better if they were to be moderately changed. This is not a top priority, but on the other hand things that will be corrected quickly.

Templates are a good idea for Content Studio to develop even more than they already have. They let the customers adjust provided suggestions instead of having to create everything from scratch. Examples of templates are image galleries, news lists and other functions often requested by users. Microsoft’s new guidelines were very similar to the already existing ones, and therefore not providing any new guidance, apart from the specific graphical ones aimed at developers. These are however, not of interest for this work and are therefore not taken into consideration.

The results from the different evaluations show that the areas the solution should be focused on are user assistance (help and documentation), cluttered workspace (window management, interface flexibility and icons) and overall consistency and feedback.

8 Recommendations

Some of the problems are due to technical issues, which somewhat affect the recommendations. At the same time many of the issues exist as a result of lack of time when developing and can easily be solved when time is available.

These recommendations encourage an overall consistency, which should be applied to any similar, undiscovered, problems that might be a part of the current version of the application. Since the future changes have to be implemented in the current system, the recommendations are made to illustrate the possible solutions. The suggestions are focused on the interaction between the user and the application, and aimed at placement of buttons and structure of dialogs and menus. Form and colours, such as gradients and other visual enhancements, are not addressed. Thus, the colours used when illustrating the suggestions more graphically are not particularly meant as part of the recommendation.

The use of windows standards seemed to appeal to most of the customers. And according to Nielsen [26] user testing has shown that users often demand that other user interfaces work like Office. When using one style most of the day, it is often wanted in other applications and screens as well. Because of this, there is no reason not to continue with the current style of the UI.

One might be sceptical towards Microsoft as a concept, but on the other hand conventions do not become conventions unless they work. Therefore there is no need to invent the wheel again, unless one is certain that one has a better idea and that the replacement is self explanatory or worth the extra time it takes to learn it [12].

The problem areas discovered during the evaluation was grouped into 3 parts as mentioned in the previous section. The most important recommendations are presented in more detail in the following text.

8.1 Cluttered Workspace

The interface is not especially cluttered, until one opens several documents and/or has a small screen. The two primary areas that can help to solve this are window management and interface flexibility.

8.1.1 Window management

When performing the heuristic evaluation the presence of the browser toolbar was seen as a bit bothering. The survey showed that some users thought of Content

Studio as part of the web and that as many users that considered Content Studio to be a separate program, considered it to be a combination between the two. To ensure that the browser environment does not cause any confusion, a way to remove the browser toolbar would be preferable.

If this is not done, the users risk confusing the functions of the browser and the program and might for instance click on “back” to undo an action or move to a different view. The result however is that the previous webpage is shown and the program is terminated. As soon as the program is shown in a separate window without browser controls the users are able to use it as any other program. The users should not have to care about whether they are working on the web or just towards the hard drive [25].

There already exists a simple way of getting rid of the visual effects of working in a browser environment. That is to view the web page in full screen mode. This can be done both through the “view” menu in the browser and by pressing F11. However, there currently does not appear to exist any way of forcing the users’ browser into this mode by using JavaScript for example. It seems as if the only way is to inform those who are bothered by the double environment to use this command to solve the problem. To further add an element showing that the application environment differs from the web, a headline such as the one present in the simple mode and in EPiServer should be added to Content Studio (see Figure 8.1). This will enhance the feeling of it being a separate program and will be helpful in normal mode, but even more when viewing the application in full screen mode.

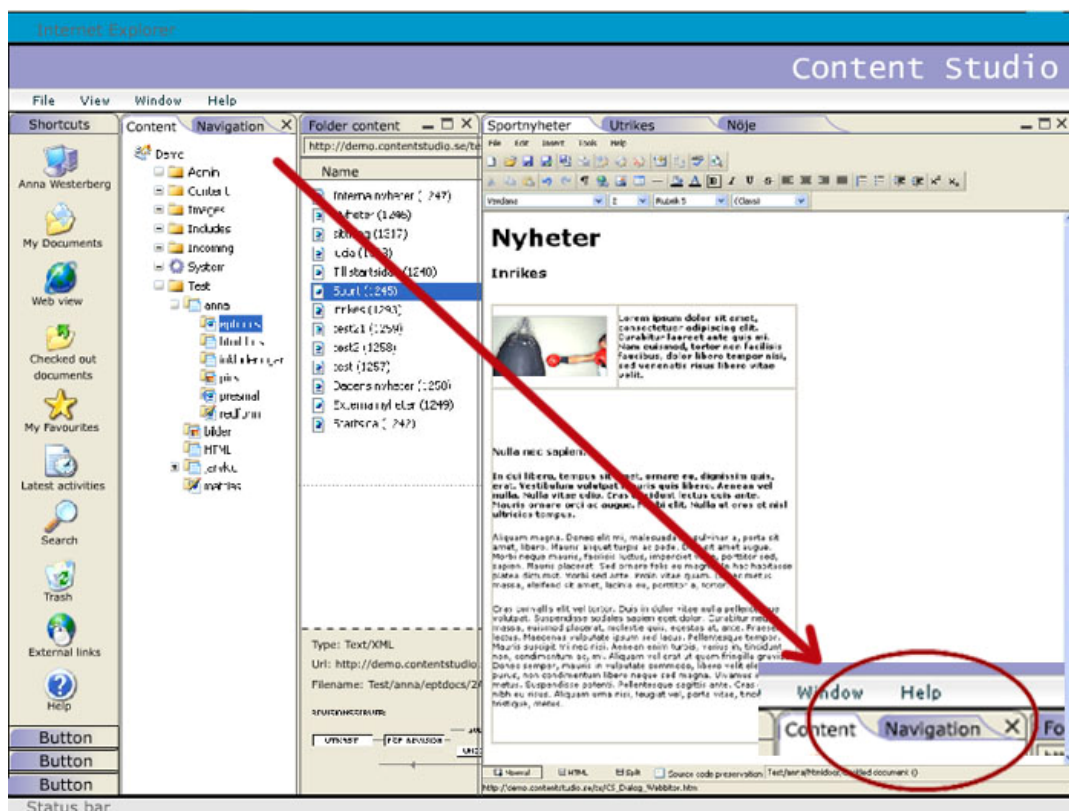


Figure 8.1. Tabs used to switch between open documents. The enlargement in the bottom right corner shows that different objects, such as the web tree, can contain more than one tab. The active tab is plain, while the other tab(s) are displayed with the affordance of a button. Each object can be manipulated and enlarged, decreased and/or closed. Regaining a closed object can be done through the view menu.

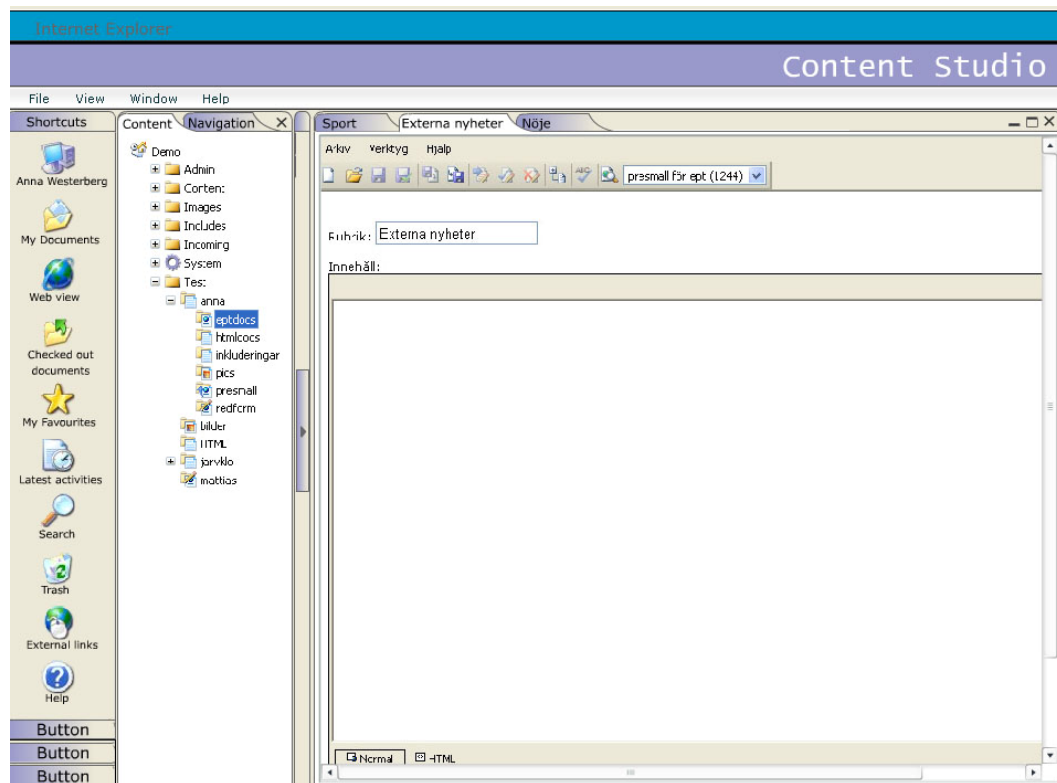


Figure 8.2. Minimizing a tab such as the folder content results in an easy accessible, clickable area, which allows the tab to be opened again.

Novice users often have difficulty with window management. They do not always realize that overlapping windows represent a three-dimensional space. As a result, when a window hides another, a user may assume it no longer exists [15]. The most preferable way of dealing with several open windows at the same time is to represent every window as a separate tab. This still shows one window at the time, but enables the user to get a clear overview and quick access to other open widows. This is used by several other programs and is a familiar way of switching between documents. Furthermore, it seems like Office 12 will make use of tabs to manage open files in Access for example.

When talking to the developers this idea was well received, but too complex to put into practise for the time being. The underlying structure is currently built on a type of windows that cannot be converted into tabs very easily. This, however, does not mean that the idea of using tabs should be discarded. One way of displaying windows through tabs are showed in Figure 8.1 and 8.2. This solution was based on the need for the users to be able to view the web tree and its corresponding file information. Without it, they have no access to created documents. The suggested solution keeps these areas, while still using tabs as a way to ease the window management, but makes it possible for the users to choose how much of each area they need to see.

Another solution, which at least could be used temporarily, is to make use of the same functionality as Word currently possesses. It contains a menu called windows, where every open document is listed (see Figure 8.3). When an item is selected, the corresponding document is placed in front of the others. To make this even more

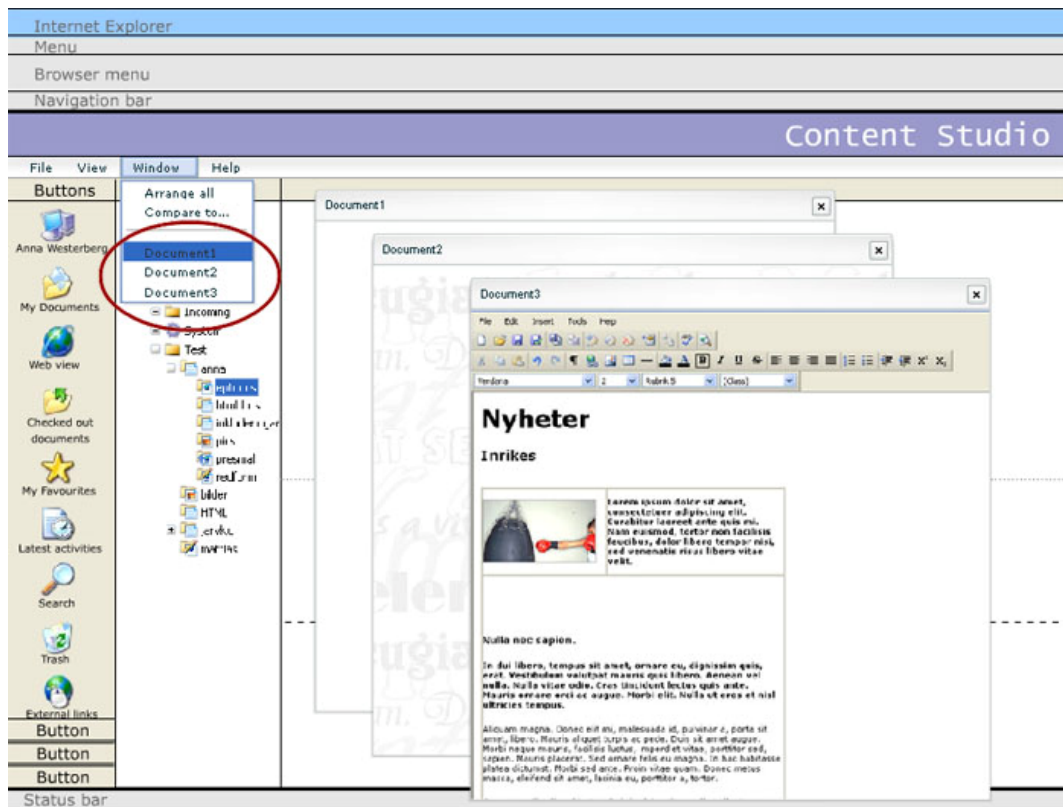


Figure 8.3. Use of a windows menu, which displays the open documents.

visible, it might not have to be hidden in a separate menu, but can be displayed as a separate field or a button. For instance, it could be included in the buttons to the left and thereby allowing the user to have an overview available at all times. At least the user should be given the possibility to change fast between different amounts and views of information instead of having every item in a separate window. If the users feel that they have an overview of the information they will experience that they are in control. [13].

8.1.2 Interface flexibility

To support the users' need for flexibility and personalization, a minimum requirement is to enable them to change the column widths. As well as users wanting more tool tips, there are some that want to be able to turn them off. This kind of personalization can also be supplied. In addition, the suggestion for window management presented above contains many possibilities for the user to choose how much information he or she wants or needs. A new Active Scripting component created during this work allows the administrator to be able to choose for instance: if a question for checking in a document should be shown when closing or if the document should be checked out automatically. Remembering to check out a document was a problem for many of the users, but this setting effectively takes care of it. This fits well into the users' needs and is an excellent way of increasing the interface flexibility. More ideas in a similar line of thought can only help improve the users' experience.

8.2 User assistance

Using any interface for the first time is a challenge for many users. Learning something new does not have to be frustrating, but often is for many people. Experienced users also have to focus to be able to learn more difficult aspects.

In most of these cases when a user gets stuck, manuals, tutorials and online help can be useful. Tutorials are most important for novice users, while support such as online help helps bridge the gap between what the users know and what they need to know and are best suited for intermittent users. Today's interactive systems are expected to assist the user by providing online help, manuals and tutorials [32]. The need for this will probably never be eliminated, no matter how good the user interface becomes. At the same time, most users avoid user manuals and prefer to learn the system by exploration [30]. The result from the survey showed that 15 of 23 had never used the online help and those who had did not rate it very high.

8.2.1 Help

Help is not something to be used to compensate for poor interface design. It should be supplied as a mean to assist the users when they need it, providing different help for different types of users. An important aspect of a help system is that it is easy to reach and easy to return from [32].

Part of the problem with Content Studio's help function may be that it is separated from the application. It takes the user out of the working context, which makes it easy for the user to lose track of what they were doing and difficult to connect the information to the encountered problem [7]. The ability to provide context sensitive help is one of the advantages of using online help as user assistance.

Tool tips are a kind of context sensitive help, well suited for additional information. One can also use cascading tool tips as presented by Constantine and Lockwood [7], which provide a secondary text that expands on request or after a second delay. According to them, guidance as opposed to help is always present in context of the work and provided when needed without the user having to take any special action. Tool tips already exist in Content Studio, but can be used more consistently throughout the interface. Many times when this assistance is not provided as a result from lack of time when developing. This is why it would be easily taken care of during further development.

The survey showed that the users anticipated that a higher degree of learning was needed to be able to use Content Studio to the fullest. However, one can shorten the learning curve. To solve the problem with Content Studio's unavailable help, a more guiding interface should be provided. Since the users mentioned that more declarative instructions were needed sometimes, a way to remedy this is to place a button in the dialog box, presenting guidance for that particular function. Figure 8.4 shows how this could look. From there a link that leads directly to the corresponding section in the online help can be provided. This will allow the users to view the explanations needed, and to be offered to follow a link to more information. Thereby the previously, maybe not even discovered, help section is placed at the users' fingertips. The context guidance can help reducing the most unnecessary and trivial calls to the support.

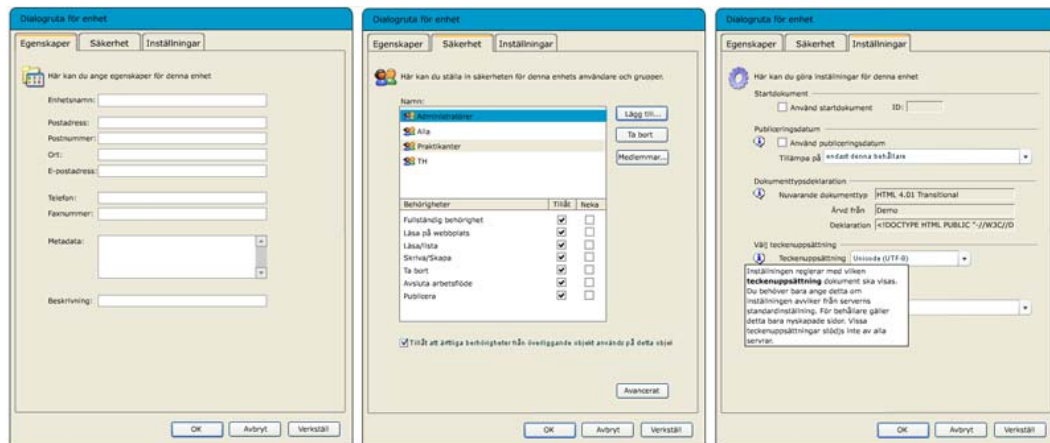


Figure 8.4. Dialog boxes for properties settings consisting of tabs, which makes it possible to follow the same format for every properties dialog. The last dialog box shows how the context guidance could be implemented.

By making the UI provide instructive interaction which guides the user to correct usage, the user perceives the interface as more complete as a consequence of having easy access to all the needed help and guidance [7].

8.2.2 Documentation

Of course documentation already exists. However, since so many of the participants in the survey mentioned a lack of documentation it is clear that there is work to be done in that area. Not only did they want more extensive documentation, they also requested information about new releases and updates. Something such as a newsletter could easily solve this. It should describe what the new function is, why it is needed and how it should be used. A list with a summary and a usage description, together with some examples, would help out a lot.

The documentation should be more easily accessed from the system. It is great to provide it as an external link, since it then can be kept open and used as a complement when working. The page one is linked to should, however, be more adjusted to fit the users' needs. The support area should for instance be more structured, visible and provide a clear overview of its contents when the user enters the online help.

8.2.3 Language

The language can be fairly technical, and might get in the way for more inexperienced users. Since Content Studio is marketed as an application that does not require any programming skills, a language that is not built on technical terms should be used. By using more comprehensible language that explains functions and components, as well as menus and dialog boxes, the users do not feel as intimidated. Every message and instruction should be presented with a friendly, positive and supportive tone so that the user can understand the content and act on the information.

For example, instead of using “_blank” for selecting a link to open in a new window, a more declarative text, such as “open in new window” should be used. As mentioned earlier, this is something that the competition has succeeded in doing and something worth considering. Sometimes, as for figure 7.5, the message depends on the system’s difficulty in knowing the unnamed documents. Still, a more comprehensible text, which at least differs between the document to be created and the existing one, can be provided. According to the company inUse [13], good communication between the user and the system will contribute to a more relaxed usage and fewer calls to the helpdesk.

8.2.4 Templates

Templates can be used in many areas; for example one can provide CSS templates for displaying images with additional graphical elements such as a shadow or a frame. The goal with any template is to help the user create results which he or she does not have knowledge of how to build. By using a prepared template this is done quickly and easily.

Templates for discussion forums, image galleries, forms, calendar and such, should be provided to help the users reach their goals without knowing how. This approach is somewhat used by EPiServer and will be incorporated in Office 12 as well. This is a good way to enable inexperienced users in particular to create good results. The development of templates also allows some control over the results, which can help the customers to produce more usable websites. Content Studio already supports the ability to insert templates and combining Active Scripting components into functions, and can therefore expand it to include galleries or other types of objects.

8.3 Overall consistency

8.3.1 Coherent look and feel

The head of the type of windows that is used in Content Studio cannot contain any other buttons than exit and help. Therefore the button for maximizing has been deliberately coded into the application in some dialog boxes. However, there may not be a need for it, since the window is resizable. As it looks now, it breaks the coherent look and feel of both this application and other windows applications. This button can be removed without it affecting the usage of the program in a major way. The removal will instead support consistency. Regarding what the menu option “create” results in, supplying options on what to create, such as a new category or an ordinary webpage, can be a solution. This is already implemented in the right-click menu and should therefore not be a problem to achieve.

Consequently placing the buttons in the same place makes the user learn where they are placed. When a pattern is created the user does not have to think consciously about it and can use the knowledge stored in the long-term memory. The user does

not have to look for the wanted button, which decreases the risk of failure and makes the user more efficient. This must be applied to the dialog boxes for properties settings, where the same buttons and functions are displayed in numerous different places from one dialog box to another.

8.3.2 Consistent feedback

To give the user continuous and evident feedback contributes to giving the user control of the system. As the guidelines described earlier, it is important that the user feels in control of the system and not vice versa.

The web tree should have underlined text and/or a shaded colour on the highlighted alternative. This is also the way Windows indicates a selectable, but not yet chosen, area in a tree structure. This also enhances the feeling of it being a link and, thus, something clickable that will provide additional information. Applying this on lists and areas with selectable items will increase the feedback of what is chosen and selectable. This includes highlighting that corresponds to the correct item, which increases the mapping between making a choice and the outcome of that choice. By doing this, it is possible that the need for selecting “check out” becomes more obvious.

Making alternatives feel like links naturally enables them to only require single clicks. The test performed on the simple mode also showed that users were confused about when to use single clicks and when to use double clicks.

Drag and drop is a fast way of carrying out tasks. However, it should never be the only way to carry out a task and should be seen as a shortcut equivalence when using a mouse [13]. This affects the current use of the web view, since pages are only viewable if they are dragged to the icon. Additional ways of viewing a document can be to select the document and then simply click on the icon for web view and to provide a web view option in the right-click menu. The drag and drop method works well once it is learned, but is not intuitive in its current presentation. By providing other ways of performing this, the user can choose the option best corresponding to his or her way of work.

When making use of this technique it is important that the pointer tells the user when something is dragged, where it can be dropped and what operation that will be applied. Any operation that will destroy the selected object should be accompanied by a confirmation by the user. Because the drag and drop operation is a dynamical operation the user needs dynamical information on what the result of their actions will be [13]. Direct manipulation, such as this operation, gives users direct feedback, they can easily determine what happened, can spot and repair problems quickly, but it may consume valuable screen space [32].

8.3.3 Mapping and visibility

The design of dialog boxes combines menu selection and form fill in to help the user make the requested selections. Instructions that describe the users’ tasks should be brief and supply an option to more help if needed by the novice user [32].

A commonly used dialog box is the one for properties. Such a dialog box is available for the main site, every unit and every category. The settings differ somewhat between the various types, but the main options are the same. The current solution

presents the different areas in the same frame, offering buttons to perform different settings. Instead, each area could be turned into a tab. This also makes it possible to remove many of the buttons leading to a single option in a new window (see Figure 8.5). By doing this the forced modality of the first window is eliminated.

If the window only contains what is relevant for the user at that moment, the user will more easily find the wanted information and the risk of choosing the wrong button is decreased. As an alternative, a list that displays the details for a selected list item can be used instead of tabs. Combining an overview of all available information and detail lessens the cognitive workload and makes it easier for the user to concentrate on the task [13].

Users normally do not mind having to scroll down windows. However, it must be obvious that more information is available. Too much space between sections can cause problems, since many users tend to overlook the scrollbar [18]. In the dialog boxes for active scripting components, this must be applied to clearly indicate that more options are available. The options should also be grouped to give a more visible structure and make it easier to find the wanted setting. By using the Gestalt theory, information can effectively be connected visually. Sometimes it is better to use this than to follow a known standard [18]. For example by breaking the standard from Windows that declares field text as left aligned, in order to follow the law of proximity.

Another side effect of grouping is that the user can handle information in chunks. Our limited short-term memory is overloaded when too many loose items are present on the screen. Humans see figures placed close to each other as a unit, which means that we see groups of symbols as belonging to each other. This can be considered in numerous places in the interface where spaces between buttons and texts can be reduced or, in some places, increased. Further increasing the icons in the web view will make it easier for the user to find and select them, as opposed to the current appearance.

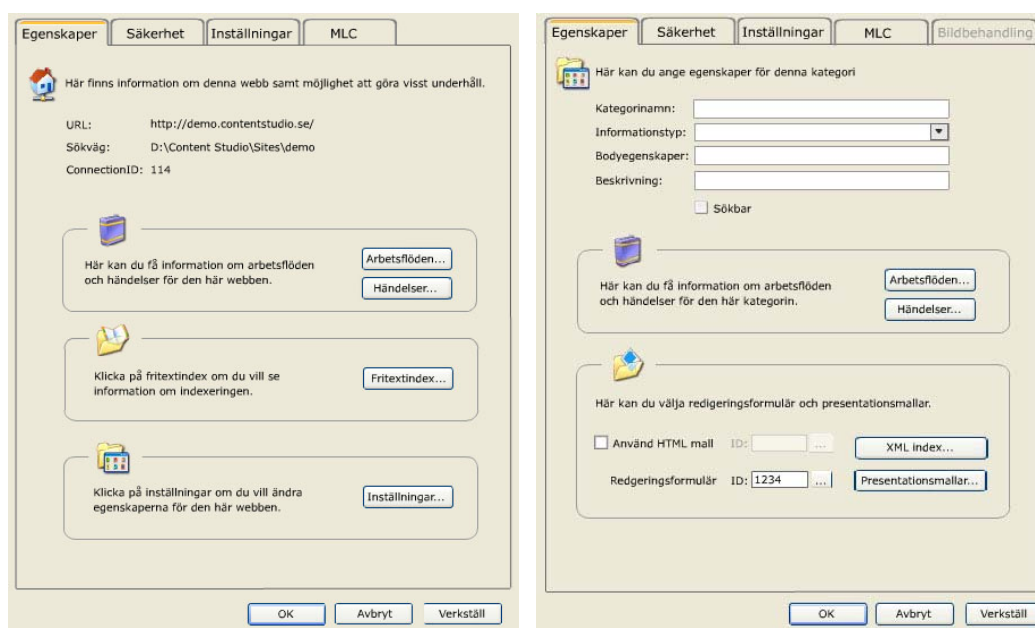


Figure 8.5. A more structured and consistent layout of the dialog boxes.

9 Summary and Conclusions

This work consisted in evaluating the interface of a web application platform. The evaluation was performed through a usability inspection and a survey, where the usability inspection consisted of an expert evaluation following specific usability principles and guidelines. The survey was then used as a complement to be able to capture the users' subjective reactions to the system. The design rationale behind applying more than one evaluation method is that if the same conclusions come from more than one approach, then the conclusion is more likely to be correct. For example, as pointed out by Brinck et al. [6], finding a problem by inspection and then observing the same problem on users can safely lead to the conclusion that it is an important problem to solve.

The paper described the most prominent deficiencies that were discovered, as well as recommendations on how to eliminate these. Some of the problems discovered in the evaluation were not possible to solve today without an extensive restructuring of the underlying infrastructure. Still, many of the problems had several possible solutions, where one could be applied immediately and another might have to wait. Once discovered, some minor problems also had rather apparent solutions.

After the data from the evaluation had been analysed the nature and severity of the problems had to be considered. Content Studio did not contain any problems so severe that they prevented the users from completing their tasks, but it had recurring problem areas, which differed in severity. Overall though, it seemed to be very appreciated by the users and ultimately it is their opinions that matters.

Ideally, a user should not need to know whether they are interacting with an application over the Internet or on their computer. However, the fact that web applications are so different from one another makes it difficult to develop a general set of accepted guidelines [40]. As Wroblewski says, it is important to address the problems associated with interface design for web applications. The most important thing is the need for a different way of thinking when it comes to these applications. The guidelines presented in the paper can assist the developers in their future work and eliminate the most minor problems in the interface.

This work has taken Content Studio a further step into an iterative design process. Although revisions previously have been made continuously they have not been as founded in a user centred design as now. Of course, more work is needed to ensure an even better design and usability quality, but a part of the introduction work is done.

10 Discussion

When forming the recommendations it was very good to have the design rationale, i.e. the reasoning that has led to the design decisions, at hand. Designers and developers may have had to make trade offs and may already be aware of the problem. They have also invested time and effort into the product, which is why a dialog with them during the process is valuable both to me as the outside expert and them as getting to know some of the outcome of the review.

Content Studio was an unexpectedly extensive system and it took a lot of time to try and explore every inch of it. Ultimately, other ways of getting to know the system was presented through the developer days, where I became educated in the system as well as in the users' needs and thoughts. My knowledge of Content Studio was therefore limited to what I had seen and customer input. If something has affected the evaluation my possible lack of knowledge in the system, might be the most prominent factor. To avoid this, the recommendations are fairly general for Content Studio as a system and contain aspects which can be applied to other parts as well.

Other possible biases are that usability inspection does not involve users and as a result an identified problem cannot be certain to create a problem for the user or that the suggested solution is not going to be worse. Here, the customer input has helped to focus on the problems in common and thereby confirming the problem areas. I am rather certain that the discovered problems are valid, since users from both sides of the knowledge spectrum mentioned some common issues. Both the interviews and the survey ranged from expert users down to more novice users and many of the problems were addressed from both ends. More problems might have been discovered if more control over the user input could have been possible. It is difficult to know how carefully the users filled out the questionnaire and what they were thinking while doing it. There is a chance that some problems are not mentioned because they are believed to be selfexplanatory and the users do not want to admit that they do not understand it.

There are always difficulties when trying to fix a design problem after an application is released, because users may adapt to, or even become dependent on, a peculiarity in the design. The foundation of Content Studio was found to be very stable and since none of the problems were so major that a complete restructuring was needed, the recommendations were concentrated on making justified changes while keeping the familiarity. Several of the complaints from the customers regarded their own website and the way they work with it. This suggests that some usability work can be needed for the productions created with Content Studio. This is where the guidelines should be applied.

Instead of asking the users immediately, a design proposal could have been made based on the heuristic evaluation. Then the users could have stated their opinions on the proposal which might have led to further changes. However, since the users were available, and the opportunity to observe them using the current

version of the system was presented, I decided to gather as much information as possible before making a suggestion. This caused the recommendation to be based on a more informed decision, thereby covering more ground and being more likely to be reasonable. The next step should be to continue testing and making use of the customers and their opinions.

10.1 Future work

A new version, Content Studio 4.5, was released during the progress of this work. For instance, the web tree was complemented with a menu structure used for navigation, but changes made in that version have not been based on the findings in this report. For future upgrades however, this report contains valuable information on problems and possible solutions as well as a set of guidelines that should be considered.

Ideally, another test would be conducted even before any changes are made. This can help deciding if the recommended changes actually will solve the usability problems that were found. The users might have a different opinion. To conduct another test will take much less time when test subjects and some material that can be used as a prototype already exist. This will also continue the iterative design process with evaluation and redesign in several steps.

After changes are made, the system should be re-evaluated [6]. This makes it possible to verify that suggested changes actually did improve the usability.

Another aspect is to keep updated on Windows future work. Their evolvement should be monitored and if Teknikhuset wants Content Studio to continue to resemble Office, it requires that they adapt to the parts well received by customers to continue into the next generation of interfaces.

Acknowledgement

The possibility to work in a close environment with the developers of Content Studio, provided a great opportunity to get the explanations needed, and a dialogue that otherwise would not have been possible. I would like to thank everyone at Teknikhuset for making me feel welcome and for letting me participate in their everyday work. I am especially grateful to Tomas Agerberg, Åke Järvklo, Erik Strandman, Linda Åhrén and Niclas Åström for helping me in various ways during my work with this paper. Thank you for your interest, encouragement and support, and for all the questions you answered.

I also want to thank my internal supervisor at Umeå University, Lena Palmquist. You have been a great help. Thank you for all your suggestions and assistance.

A special thanks to the participating end users who provided me with excellent information and their personal point of view. Your opinions were invaluable.

Finally, I am grateful to my friends and family, and especially Robban, for always being there for me. Thank you.

References

- [1] Barnum, C. M. (2002) *Usability testing and research*. Pearson Education, Inc.
- [2] Berkun, Scott. (1999). *How to avoid foolish consistency*. The human factor, Microsoft Corporation.
- [3] Berkun, Scott. (1999). *The importance of simplicity: create ease of use without losing power*. The human factor, Microsoft Corporation.
- [4] Berkun, Scott. (2000). *Fitts's UI Law applied to the web*. The human factor, Microsoft Corporation.
- [5] Boiko, B. (2002). *Content Management – the bible*. Hungry Minds, New York.
- [6] Brinck, T., Gergle, D. and Wood, S. D. (2002) *Usability for the web: designing websites that work*. Morgan Kauffman, San Francisco, CA.
- [7] Constantine, L. L., Lockwood, L. A. D. (2002). *Instructive interaction: Making innovative interfaces self-teaching*. <http://www.foruse.com/articles/instructive.pdf>, last visited 051202.
- [8] Dumas, J. S. and Redish, J. C. (1999). *A practical guide to usability testing*. Intellect Books, Exeter, UK.
- [9] Fatwire Software (2005). *Content management glossary*. http://www.fatwire.com/it/services/C_cm glossary.html, last visited 050921.
- [10] Gulliksen, J. and Göransson, B. (2002). *Användarcentrerad systemdesign*. Studentlitteratur, Lund.
- [11] Kaasgaard, K. (2000). *Software Design and Usability*. Copenhagen Business School Press.
- [12] Krug, S. (2006). *Don't make me think*. New Riders, Berkeley, CA.
- [13] InUse AB. (2004). *Guidelines for GUI design*. <http://www.inuse.se>, last visited 051102.
- [14] McCracken, H. (2005). *Microsoft Office 12's Radical New Look*, <http://blogs.pcworld.com/techlog/archives/000900.html>, last visited 050915.
- [15] Microsoft Corporation (ed). (2004). *The windows user experience: Official Guidelines for User Interface Developers and Designers*. Microsoft Press.
- [16] Microsoft Corporation. (2005). *The future of Microsoft Office*. <http://www.microsoft.com/office/preview/default.mspx>, last visited 050915.
- [17] Microsoft Corporation. (2005). *Windows Vista UX Guidelines*. <http://www.microsoft.com/downloads>, last visited 050920.
- [18] Molich, R. (2002). *Webdesign med fokus på användbarhet*. Studentlitteratur, Lund.
- [19] Nathan, M. (2004). *X-infoModel: Glossary*. <http://www.xinfomodel.com/glossary.html>, last visited 050921.
- [20] Nielsen, J. (1992). *Finding usability problems through heuristic evaluation*. Proceedings ACM CHI'92 Conference, Monterey, CA, May 3-7: pp. 373-380.
- [21] Nielsen, J. (1993). *Usability engineering*. Academic Press, San Diego, CA.

- [22] Nielsen, J. and Mack, R. L. (1994). *Usability Inspection methods*, John Wiley & Sons Inc.
- [23] Nielsen, J. (1997). *The difference between web design and GUI design*. Alert box for May 1, <http://www.useit.com/alertbox/9705a.html>, last visited 051011.
- [24] Nielsen, J. (1999). *Intranet Portals: The Corporate Information Infrastructure*. Alert box for April 4, <http://www.useit.com/alertbox/990404.html>, last visited 051103.
- [25] Nielsen, J. (2000). *Designing web usability: The Practice of Simplicity*. New Riders Publishing, Indianapolis.
- [26] Nielsen, J. (2005). *R.I.P. WYSIWYG*. Alert box for October 10, <http://www.useit.com/alertbox/wysiwyg.html>, last visited 051011.
- [27] Norman, D. (1988). *The Design of everyday things*. MIT Press, London.
- [28] Norman, D. (2000). *The Rise of Weblications*. Keynote address presented at the User Experience World Tour, Chicago, IL.
- [29] Ottersten, I. and Berndtsson, J. (2002). *Användbarhet i praktiken*. Studentlitteratur, Lund.
- [30] Preece, J., Rogers, Y. and Sharp, H. (2002). *Interaction Design: beyond human-computer interaction*. Wiley & Sons Inc.
- [31] Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. Reading, MA: Addison-Wesley Publishing Co.
- [32] Shneiderman, B. and Plaisant, C. (2005). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education inc. Addison-Wesley. 4th edition.
- [33] Söderström, J. (2001). *Vad är interaktionsdesign?* <http://kornet.nu/iadesign.shtml>, last visited 051201.
- [34] Teknikhuset AB (2005). *Content Studio*, <http://www.contentstudio.se>, last visited 051207.
- [35] Teknikhuset AB. (2005). *Teknikhuset*, <http://www.teknikhuset.se>, last visited 050822.
- [36] Tognazzini, B. (2000). *OS X: A First Look*. AskTog, <http://www.asktog.com/columns/034OSX-FirstLook.html>, last visited 051216.
- [37] Usability First. (2005). *Introduction to usability*. <http://www.usabilityfirst.com>, last visited 060104.
- [38] Vasont Systems. (2004). *Multilingual XML Content Management for Multi-Channel Publishing*, <http://www.vasont.com/gloss.html>, last visited 050921.
- [39] White, M. (2005). *The Content Management Handbook*. Facet Publishing, London.
- [40] Wroblewski, L. and Rantanen, Esa M. (2001). *Design Considerations for Web-based Applications*. Human Factors and Ergonomics Society's Annual Meeting proceedings.

Appendix A

The heuristic evaluation resulted in a list of various types of advantages and disadvantages in Content Studio. These were classified and described more thoroughly in section 7.1. The original set is presented below.

List of problems found during the heuristic evaluation

- Some of the checkboxes in different dialogues cannot be checked even though they are white and thus indicating that they are active compared to the grey ones.
- Tool tips are inconsistent compared to Word and compared to each other.
- The back-button in the web browser toolbar can sometimes work as if it is a part of the program, but it does not belong to it, since a back and forward button also exist in the web view. Because of the web browser there also exist double sets of "file", "view" and "help", which might cause confusion.
- Single-click and double-click are mixed, sometimes following web standards and sometimes software standards.
- When choosing file-create it is difficult to understand what is created. The icon to expand this window lies beneath the exit button, which is not how it is done in Windows.
- Icons in the dialog for active scripting components are placed in a different order compared to the ordinary view.
- The same buttons in dialog boxes for units and categories are also placed in a random order. The amount of buttons has resulted in two rows of buttons, which makes it difficult to get an overview of.
- Icons for viewing folders in different ways do not have tooltips. If the users are not used to the standard, they will not help.
- The use of popup menus reduces the clutter in the UI and is only showed when the users request them, which is good. Although, several of the dialog boxes for Active Scripting components are very long and sometimes ending a section just before the window edge, which can cause users to believe that no more options are available.
- Dialog boxes that leads to the opening of other dialog boxes becomes modal and thereby prevents the user from looking at the documents underneath.
- To exit the application does not result in any feedback.
- For "insert image" or "open" a preview option is available, but no indication that the text is selectable is given. Instead, preview is easily mistaken for the already open frame.

- The web tree does not indicate which of the items that is aimed at when the mouse pointer moves over it.
- When connecting a presentation template to a document, "add" is required. The user might wait for alternatives to be shown as when selecting an edit template.
- Check out should make anything not selectable unavailable to more clearly indicate the need to click it.
- Even though the menu items are greyed out, the icons next to the item text is not. The entire item and not just the text should be grey.
- When navigating in the file system, for instance when creating a new document, the hierarchy expands folders which do not contain other folders and therefore cannot create anything new.
- Revision control supports exploration in an excellent way, since the user always can retrieve an old version. However, restore version should be able to do without having to enter editing mode. An undo button is also available, making small changes easy to reverse.
- The easy mode helps novice users to edit information in a simple way.
- Not every shortcut is available as a menu option and may only be accessed by a right click or an icon.
- Some messages do not provide enough information for the user to make an informed choice.
- Most explanations for fields in forms are clear, which is good.
- "_blank" is used to select a link to be opened in a new window. This can be difficult to understand and remember for inexperienced users.
- The only way to look at a page in the web view is to drag it to the icon. The drag and drop function is not obvious and to drop it on the correct spot, one has to cross an area where the document either cannot be dropped or can be inserted into a folder.
- Drag and drop is also supported for example to insert text from other documents or pages, which is good.
- When using a small screen, the workspace gets very cluttered. Windows on top of each other makes it difficult to locate the ones at the bottom. Together they hide the web tree and the document information which is needed to open other windows.
- When requesting help, the user is forced to log in instead of being linked directly to the area in question.

Appendix B

This is a copy of the content in the questionnaire. The text fields for free comments are removed, but the questions where the users were asked to grade their opinion on a scale are presented here as (statement 1) 1 2 3 4 5 6 7 (statement 2). The questions lacking a scale are therefore the free comments. Every scale had an option where the user could choose to be uncertain. This is not showed here. The survey was conducted in Swedish and presented in its original form as follows.

Enkät för användares tillfredsställelse med interaktionen

Dessa frågor är till för att hjälpa mig identifiera för- och nackdelar i Content Studios användargränssnitt. Det är inte en utvärdering av dina kunskaper eller arbetsuppgifter utan ett tillfälle för dig att påverka kommande utformning av systemet. Frågorna ska hjälpa mig förstå vad du som användare tycker om produkten baserat på de uppgifter du är van att utföra.

Frågorna är indelade i olika områden. Markera det alternativ som bäst passar dina intryck av Content Studio. För varje område finns ett fält för kommentarer, där du gärna får förtydliga, ge synpunkter eller förslag.

Enkäten tar ungefär 15 minuter att fylla i. Alla svar är givetvis anonyma.

1. Erfarenhet

Vad arbetar du med?

Hur länge har du arbetat med Content Studio?

- ☐ mindre än 3 månader
- ☐ 3 månader - 1 år
- ☐ mer än 1 år men inte mer än 3 år
- ☐ mer än 3 år

Hur mycket tid per vecka lägger du ner i genomsnitt på arbete med Content Studio?

- ☐ mindre än 1h
- ☐ 1-5h
- ☐ 6-10h
- ☐ 11-20h
- ☐ mer än 20h

Hur ser ditt vardagliga arbete ut i Content Studio?

Har du använt liknande system förut?

- ☐ Ja
- ☐ Nej
- ☐ Vet ej

Om ja, vilket?

Vilka andra typer av datorkunskaper har du?

Kommentarer:

2. Helhetsintryck

Att använda Content Studio är: (väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Mitt allmänna intryck av Content Studio är att det är:

(Hemskt) 1 2 3 4 5 6 7 (underbart)

(Frustrerande) 1 2 3 4 5 6 7 (tillfredsställande)

(Tråkigt) 1 2 3 4 5 6 7 (stimulerande)

(Otillräckligt) 1 2 3 4 5 6 7 (tillräckligt)

(Stelt) 1 2 3 4 5 6 7 (flexibelt)

Kommentarer:

3. Information

Tecken och färger gör information: (Svår att läsa) 1 2 3 4 5 6 7 (lätt att läsa)

Skärmlayouten är: (Ologisk) 1 2 3 4 5 6 7 (logisk)

Mängden information som kan visas på skärmen är: (Otillräcklig) 1 2 3 4 5 6 7 (tillräcklig)

Språket i systemet är: (inkonsekvent) 1 2 3 4 5 6 7 (konsekvent)

Dataterminologi används: (För ofta) 1 2 3 4 5 6 7 (lämpligt)

Meddelanden som uppenbarar sig på skärmen är: (Inkonsekventa) 1 2 3 4 5 6 7 (konsekventa)

Instruktioner för kommandon eller funktioner är: (Förvirrande) 1 2 3 4 5 6 7 (tydliga)

Att förstå instruktioner i dialogrutor är: (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Felmeddelanden förtydligar problemet: (Aldrig) 1 2 3 4 5 6 7 (alltid)

Formuleringar av meddelanden är: (otrevliga) 1 2 3 4 5 6 7 (trevliga)

Kommentarer:

4. Lärande

Att komma igång är: (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Att lära sig manövrera Content Studio är: (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Att lära sig avancerade finesser är: (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Utforskning av funktioner känns

a) (avskräckande) 1 2 3 4 5 6 7 (uppmuntrande)

b) (Riskabelt) 1 2 3 4 5 6 7 (säkert)

Att upptäcka nya funktioner är (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Att rätta till misstag är (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Förmåga att ångra operationer är (otillräcklig) 1 2 3 4 5 6 7 (tillräcklig)

Operationer leder till ett förutsägbart resultat (aldrig) 1 2 3 4 5 6 7 (alltid)

Man kan använda funktioner (med svårighet) 1 2 3 4 5 6 7 (med lätthet)

Man kan använda genvägar (med svårighet) 1 2 3 4 5 6 7 (med lätthet)

Att komma ihåg namn och användning av kommandon och funktioner är
(Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Att hitta önskade menyalternativ är (Väldigt svårt) 1 2 3 4 5 (väldigt lätt)

Kommentarer:

5. Hjälp

Brukar du använda online-hjälpen för att klara av att utföra uppgifter?

Terminologin som används är: (Förvirrande) 1 2 3 4 5 6 7 (tydlig)

Online-hjälpen är lätt att förstå: (aldrig) 1 2 3 4 5 6 7 (alltid)

Mängden hjälp som ges online är: (otillräcklig) 1 2 3 4 5 6 7 (tillräcklig)

Att hitta en lösning på ett problem genom att använda online-hjälpen görs:
(med svårighet) 1 2 3 4 5 6 7 (med lätthet)

Online-hjälpen är: (värdelös) 1 2 3 4 5 6 7 (användbar)

Kommentarer:

6. Synpunkter

Hur upplever du Content Studio?

- Som ett eget program
- Som en del av webben
- Kombination
- Annat
- vet ej

Vad gillar du mest med Content Studio?

Vad gillar du minst med Content Studio?

Skulle du rekommendera Content Studio till andra? Varför?

Är det något annat du vill kommentera?

Appendix C

The following diagrams show the different areas from the questionnaire, put together based on the users' experience. For each question a mean for the answers for each user type was calculated and this is represented in the y-axis. Several of the questions showed that expert users tended to rate the interface higher than the novice users.

